319-CD-002-002

EOSDIS Core System Project

Science Data Processing Segment (SDPS) Integration and Test Plan for the ECS Project Volume 1: IR1

Final

March 1995

Hughes Applied Information Systems
Landover, Maryland

SDPS Integration and Test Plan for the ECS Project Volume 1: IR-1

March 1995

Prepared Under Contract NAS5-60000
CDRL Item 054

SUBMITTED BY

Stephen Fox	3/20/95
Stephen A. Fox, SDP Segment Manager EOSDIS Core System Project	Date

Hughes Applied Information Systems
Landover, Maryland

This page intentionally left blank.

Preface

This document is a formal contract deliverable with an approval code 1. It requires Government review and approval prior to acceptance and use. Changes to this document also require Government approval prior to acceptance and use. Changes to this document shall be made by document change notice (DCN) or by complete revision.

Once approved, this document shall be under ECS Project Configuration Control. Any questions or proposed changes should be addressed to:

Data Management Office
The ECS Project Office
Hughes Applied Information Systems
1616 McCormick Dr.
Landover, MD 20785

This page intentionally left blank.

Abstract

This document specifies the Test Plan (IR-1) for the Science Data Processing Segment (SDPS) of the ECS Project.

Keywords: Integration, Test, I&T, Build, Thread, Interim Release One (IR-1), SDPS, ECS

This page intentionally left blank.

Change Information Page

List of Effective Pages	
Page Number	**Issue**
Title	Final
iii through xii	Final
1-1 through 1-2	Final
2-1 through 2-2	Final
3-1 through 3-10	Final
4-1 through 4-44	Final
A-1 through A-2	Final
B-1 through B-8	Final
AB-1 through AB-4	Final

Document History			
Document Number	**Status/Issue**	**Publication Date**	**CCR Number**
319-CD-002-001	Review Copy	February 1995	
319-CD-002-002	Final	March 1995	

This page intentionally left blank.

Contents

Preface

Abstract

1. Introduction

2. Related Documentation

3. SDPS Integration and Test Overview

Figures

Tables

This page intentionally left blank.

1. Introduction

1.1 Identification

This document is submitted as required by CDRL item 054, whose requirements are specified in DID 319/DV1, as a required deliverable under the Earth Observing System (EOS) Data and Information System (EOSDIS) Core System (ECS), Contract (NAS5-60000).

1.2 Scope

This document defines the plan for integration, test, and verification of the Science and Data Processing Segment (SDPS) for Interim Release One (IR1). It is one of three segment test plans required to test ECS at the segment level. There is a separate test plan for the Flight Operations Segment (FOS) and the Communications and Systems Management Segment (CSMS). The SDPS Integration and Test Plan applies only to segment level verification activities. This plan includes verifying that the ECS complies with the SDPS Level 4 functional requirements, and the ECS design specifications. The roles and activities of the Science Data Processing Segment Integration and Test (SDPS I&T) Organization are described and schedules for performing SDPS I&T activities are included.

This document reflects the Technical Baseline submitted via contract correspondence number ECS 194-00343.

1.3 Purpose

This Segment Integration and Test Plan describes the test, review, and analysis effort to be conducted by the SDPS I&T organization for the SDP Segment. This document presents the overall processes and activities associated with verifying the SDP Segment during the segment integration and test phase of SDPS development. This test plan provides an outline of the activities to be performed for SDPS I&T, and is later used to prepare test procedures which provide more detailed instructions for verification of the SDPS. It delineates the roles and responsibilities of each organization associated with the segment integration and test activities.

1.4 Status and Schedule

Volume 1 of the Segment Integration and Test Plan (DID # 319/DV1) is delivered at Release A PDR. Volume 1 defines the process for integration and test of SDPS software components and subsystems for IR1. Also at PDR, Volume 2 of the Segment Integration and Test Plan (DID # 319/DV1) defines the process for SDPS I&T of software components and subsystems for Release A. For both IR1 and Release A documents, test cases are described at a summary level identifying test objectives, inputs, outputs, and success criteria. Test databases and test tools needed for each test are identified. Corresponding documents for Releases B, C, and D will be provided at appropriate IDRs.

This submittal of Volume 1 and 2 of DID 319/DV1 meets the milestone specified in the Contract Data Requirements List (CDRL) of NASA contract NAS5060000. It is anticipated that this submittal will be reviewed during the appropriate segment- or system-level Preliminary Design Review (PDR), and that subsequent changes to the document will be incorporated into a resubmittal according to a schedule mutually agreed to by Goddard Space Flight Center (GSFC) and ECS.

1.5 Organization

This document is organized into four sections and three appendices:

Section 1 Introduction, contains the identification, scope, purpose and objectives, status and schedule, and document organization.

Section 2 Related Documents, provides a bibliography of parent, applicable and reference documents for the SDPS Integration and Test Document.

Section 3 SDPS Integration and Test Overview, describes the process used to integrate and test the SDPS and subsystems.

Section 4 SDPS IR1 Test Descriptions, describes the specific segment level thread and build tests, which will be used to verify the functionality of the SDPS.

Appendix A Test Tool Requirements, contains a list and brief description of the test tools needed for SDPS Integration and Test Tools.

Appendix B Verification Traceability Matrix, contains the requirements traceability matrix, mapping test cases to SDPS Level 4 requirements.

Appendix C Acronyms, contains a list of acronyms included in this document.

2. Related Documentation

2.1 Parent Documents

The parent document is the document from which this SDPS Integration and Test Plan (IR1) scope and content are derived.

101-101-MG1-001	Project Management Plan for the EOSDIS Core System
107-CD-001-007	Level 1 Master Schedule for the ECS Project
304-CD-002-001	Science Data Processing Segment (SDPS) Requirements Specification for the ECS Project, Review Copy
194-401-VE1-002	Verification Plan for the ECS Project, Final
194-501-PA1-001	Earth Observing System (EOS) Performance Assurance Requirements for the EOSDIS Core System (ECS)
423-41-01	EOSDIS Core System Statement of Work
423-41-03	EOSDIS Core System Contract Data Requirements List Document

2.2 Applicable Documents

The following documents are referenced within this SDPS Integration and Test Plan (IR1), or are directly applicable, or contain policies or other directive matters that are binding upon the content of this volume.

194-201-SE1-001	Systems Engineering Plan for the ECS Project
194-207-SE1-001	System Design Specification for the ECS Project
194-219-SE1-018	Interface Requirements Document Between EOSDIS Core System (ECS) and Tropical Rainfall Measuring Mission (TRMM) Ground System
194-301-DV1-002	System Implementation Plan for the ECS Project
305-CD-002-001	Science Data Processing Segment (SDPS) Design Specification for the ECS Project, Review Copy
423-41-02	Functional and Performance Requirements Specification [F&PRS] for the Earth Observing System Data and Information System (EOSDIS) Core System, Revision A

2.3 Information Documents

2.3.1 Information Documents Referenced

The following documents are referenced herein and, amplify or clarify the information presented in this document. These documents are not binding on the content of the ECS SDPS Integration and Test Plan (IR1).

194-102-MG1-001	Configuration Management Plan for the ECS Project
193-103-MG3-001	Configuration Management Procedures for the ECS Project, 10/93
402-CD-002-002	System Integration and Test Plan for the ECS Project, Volume 2: Release IR-1
409-CD-001-003	Overall System Acceptance Test Plan for the ECS Project
194-415-VE1-003	Acceptance Testing Management Plan for the ECS Project, Final

2.3.2 Information Documents Not Referenced

The following documents, although not referenced herein and/or not directly applicable, do amplify or clarify the information presented in this document. These documents are not binding on the content of the SDPS Integration and Test Plan (IR1).

104-CD-001-003	Data Management Plan for the ECS Project
193-105-MG3-001	Data Management Procedures for the ECS Project

3. SDPS Integration and Test Overview

This section contains an overview of the approach taken by the Science Data Processing Segment Integration and Test organization to ensure complete and thorough testing at the segment level. Included is information concerning the SDPS I&T environment, schedules and verification activities and responsibilities.

3.1 SDPS I&T and the ECS Environment

3.1.1 SDPS Functional Overview

ECS is comprised of three segments: the Flight Operations Segment (FOS), Communications and System Management Segment (CSMS), and Science Data Processing Segment (SPDS). Each of these segments are decomposed into subsystems and the subsystems are composed of Computer Software Configuration Items (CSCIs). This document will test the design and implementation of the SDPS CSCI's and their integration into SDPS subsystems.

The CSCI's for SDPS at IR1 are listed in Table 3-1. Included are CSCI names and SDPS subsystems. A short description of each CSCI is given below.

Table 3-1. SDPS IR1 CSCIs

CSCI	Id	Subsystem
Processing	DPS-PRONG	Processing
Algorithm I&T	DPS-AITTL	Processing
Ingest	INS-INGST	Ingest

<u>Processing CSCI</u>–The Processing CSCI provides functionality for managing, queuing and executing processing requests on the processing resources at a DAAC site.

<u>Algorithm I&T CSCI</u>–The Algorithm Integration and Test CSCI facilitates the integration of science processing algorithms and user methods into the operational environment of a DAAC.

<u>Ingest Services CSCI</u>–The Ingest CSCI provides the ability to accept, validate, and place data in a holding area for temporary storage to support TRMM interface testing. Data for IR1 includes TRMM data, received from TSDIS and SDPF, and NOAA data, received from NESDIS and GDAO.

3.1.2 SDPS I&T Relationship to other Test Groups

The SDPS I&T group is responsible for integration and test at the segment level. This includes acceptance of software components delivered by the development team, upon completion of unit testing. The SDPS I&T organization is responsible for the integration of these components into segment subsystems and complete and thorough testing of the integrated software and recording and reporting of any problems encountered during testing. Documentation for the SDPS I&T organization includes the SDPS I&T Plan, SDPS I&T Test Procedures and Test Report documentation. Integrated software is tested against Level 4 requirements documented in the Segment/Element Requirements Specification.

Upon completion of SDPS testing, the software is delivered to the ECS System I&T organization. This group is responsible for integration and test at the system level. This includes acceptance of all segment software including FOS, CSMS, and SDPS segments. Testing is done to confirm compliance to Level 3 requirements documented in the Functional and Performance Requirements Specification (423-41-02) and the System Design Specification (194-207-SE1-001).

The SDPS I&T group interacts with and supports other ECS and independent test organizations. This includes the Independent Acceptance Test Organization (IATO) and the EOSDIS Independent Verification and Validation (IV&V) Contractor. The IATO monitors segment tests and identifies any Level 3 requirements that can be verified through analysis of segment test results. The IV&V contractor monitors ECS verification activities.

SDPS I&T includes segment testing of SCF version of the SDP Toolkits. This testing is documented in the SCF Toolkit Segment/Element Integration & Test Notebooks.

3.2 SDPS I&T Testing Approach

The SDPS I&T approach involves the incremental integration of software components. The following sections discuss the SDPS I&T approach for IR1.

3.2.1 Segment I&T Functional Testing

SDPS I&T tests will integrate and verify SDPS CSCI functionality on an incremental basis. As software and hardware items complete unit testing, the SDPS I&T organization incrementally assembles lower-level functionality into progressively higher levels until ultimately a segment is completely integrated and tested. SDPS testing is based on the concept of functional "threads" and "builds" (see section 3.2.2). Thread/Build testing verifies Level 4 functional and performance requirements.

3.2.2 SDPS Thread/Build Methodology

The thread/build concept, which is based on the incremental aggregation of functions, is used to plan SDPS I&T activities. An SDPS thread is the set of components (software, hardware and data) and operational procedures that implement a function or set of related functions at the segment level. Threads are tested individually to facilitate Level 4 requirements verification and to isolate software problems. A build is an assemblage of threads to produce a gradual buildup of segment capabilities. This orderly progression of combining lower level software and/or hardware items to form higher level items with broader capability is the basis of SDPS integration. SPDS builds are combined with other SDPS builds and threads to produce higher level builds. Verification of threads and builds is accomplished at progressively higher and higher levels as the SDPS software is assembled for each release.

SPDS thread/build diagrams are developed for each Release. The thread/build diagram for IR1 is presented in Figure 3-1. Threads and builds are defined by examining SDPS CSCI's, Level 4 requirements and segment design specifications. The SDPS I&T organization with support from the SDPS development community, logically groups the SDPS into functional categories divided along noticeable boundaries. These categories are the basis for SDPS threads. Threads are combined to define SDPS builds. Builds include several integrated thread functions. The thread/build diagram for each SDPS release acts as a catalyst for definition of SDPS test case definition. From each build and thread on the diagram, test cases are developed. These test cases provide the basis for development of step-by-step test instruction to be documented as SDPS test procedures. The SDPS test procedures are used during test execution.

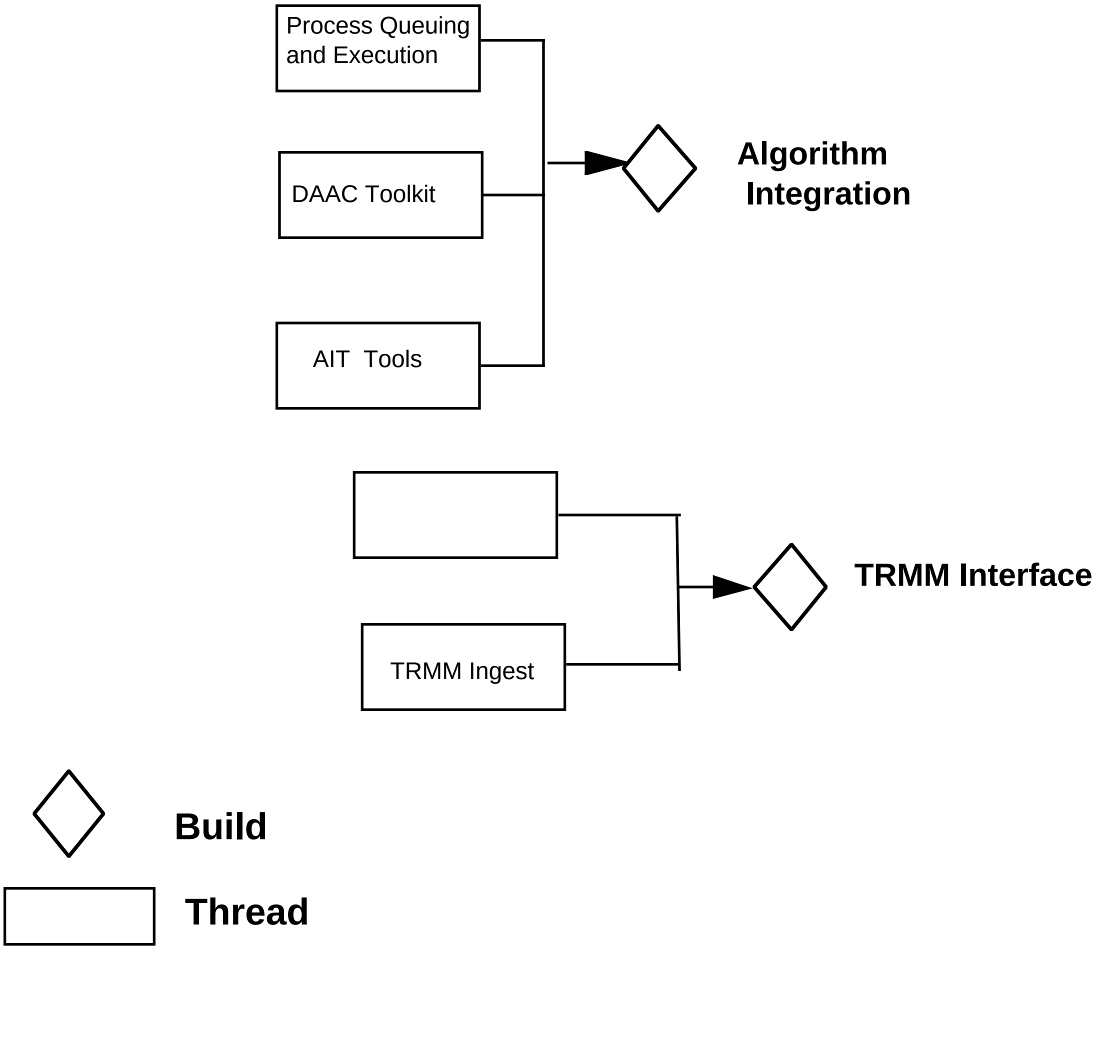

Figure 3-1. IR1 SDPS Thread/Build Diagram

3.3 SDPS I&T Test Verification

The following sections define responsibilities and activities of the SDPS I&T organization. SDPS I&T verification includes definition of verification methods, post test analysis, regression testing, and verification resources.

3.3.1 Verification Methods

The four verifications methods used for SDPS I&T include: inspection, analysis, demonstration, and test.

These are defined in the ECS Verification Plan (194-401-VE1-002).

- Inspection–The visual, manual examination of the verification item and comparison to the applicable requirement or other compliance documentation, such as engineering drawings.
- Analysis–Technical or mathematical evaluation based on calculation, interpolation, or other analytical methods.
- Demonstration–Observation of the functional operation of the verification item in a controlled environment to yield qualitative results without the use of elaborate instrumentation, procedure, or special test equipment.
- Test–A procedure or action taken to determine under real or simulated conditions the capabilities, limitations, characteristics, effectiveness, reliability, or suitability of a material, device, system, or method.

Each SDPS Level 4 requirement is tested by one or more of these methods.

3.3.2 Post Test Analysis

Post-test analysis includes data reduction and comparison of actual results against expected results. Any post test analysis required for SDPS I&T will be performed by the SDPS I&T organization with support from the user and development communities if appropriate and necessary. Methods for performing post-test analysis will be documented in the Segment/Element Integration and Test Procedures on a test by test basis. Post-test analysis will be documented in SDPS I&T Test Report. Test data, data logs, event logs and any other test output required for post test analysis is captured and stored under Configuration Management (CM) control.

3.3.3 Regression Testing

Regression testing is supplemental testing performed at any time upon any thread or build testing during SDPS I&T testing to ensure that existing software is not adversely affected by modified or new software. SDPS I&T members are responsible for the planning, documenting, executing and reporting of all regression testing. Automated test tools are used whenever possible to conduct regression testing by the SDPS I&T organization. This ensures that regression tests duplicate initial test procedures.

Regression testing is executed when new versions of software are delivered by the unit test team due to software changes or operational enhancements. Analysis is performed by the SDPS I&T team to determine the extent of regression testing required. The thread/build approach lends itself to regression testing. Builds, regression test components tested in previous threads or builds. Regression test analysis determines if the software changes or operational enhancements require regression testing all prior threads and build or a subset. Therefore no special indication for this type of regression testing is included on the thread/build diagram.

SDPS I&T is responsible for reporting any discrepancies encountered during segment regression testing. Discrepancies resulting from any other level of testing (i.e. System Test, Acceptance Test) which results in modification a the unit level, will be regression tested at the segment level by the SDPS I&T organization.

3.3.4 Verification Resources

The following paragraphs in this section introduce and identify the resources necessary to accomplish SDPS I&T. Included are identification of test location and hardware and software configurations. Also discussed is the use of automated test tools, discrepancy reporting, and the role of CM in SDPS I&T.

3.3.4.1 Testing Facilities

The ECS Development Facility (EDF) located at the ECS facility in Landover , Md. has been designated as the testing facility for SDPS I&T. ECS is solely responsible for the test environment. This includes installation, initial checkout and startup, upgrades/version control, access control, and maintenance.

3.3.4.1.1 Hardware Items

Upon IR1 test execution the EDF hardware configuration is as shown in Figure 3-2.

A description of the hardware configuration includes:

- The AI&T Processing Host is a SGI Power Challenge SMP Class server with a machine attached Redundant Arrays of Inexpensive Disks (RAID).
- The Toolkit/Algorithm Workstations are multi-vendor platforms for AI&T support and Toolkit testing.
- The Ingest Workstation has a machine attached RAID and a router connects to NASCOM Operational Local Area Network (NOLAN).
- The Processing Management Workstations are used to support process queuing and execution on the AI&T host.

- The Bulletin Board Server is connected to a router to server as a "firewall" to Internet. It provides a connection to simulate Algorithm Package delivery from a Science Computing Facilities (SCFs).
- The router to STL is used to "pull-in" the necessary resources to simulate the large DAACs.

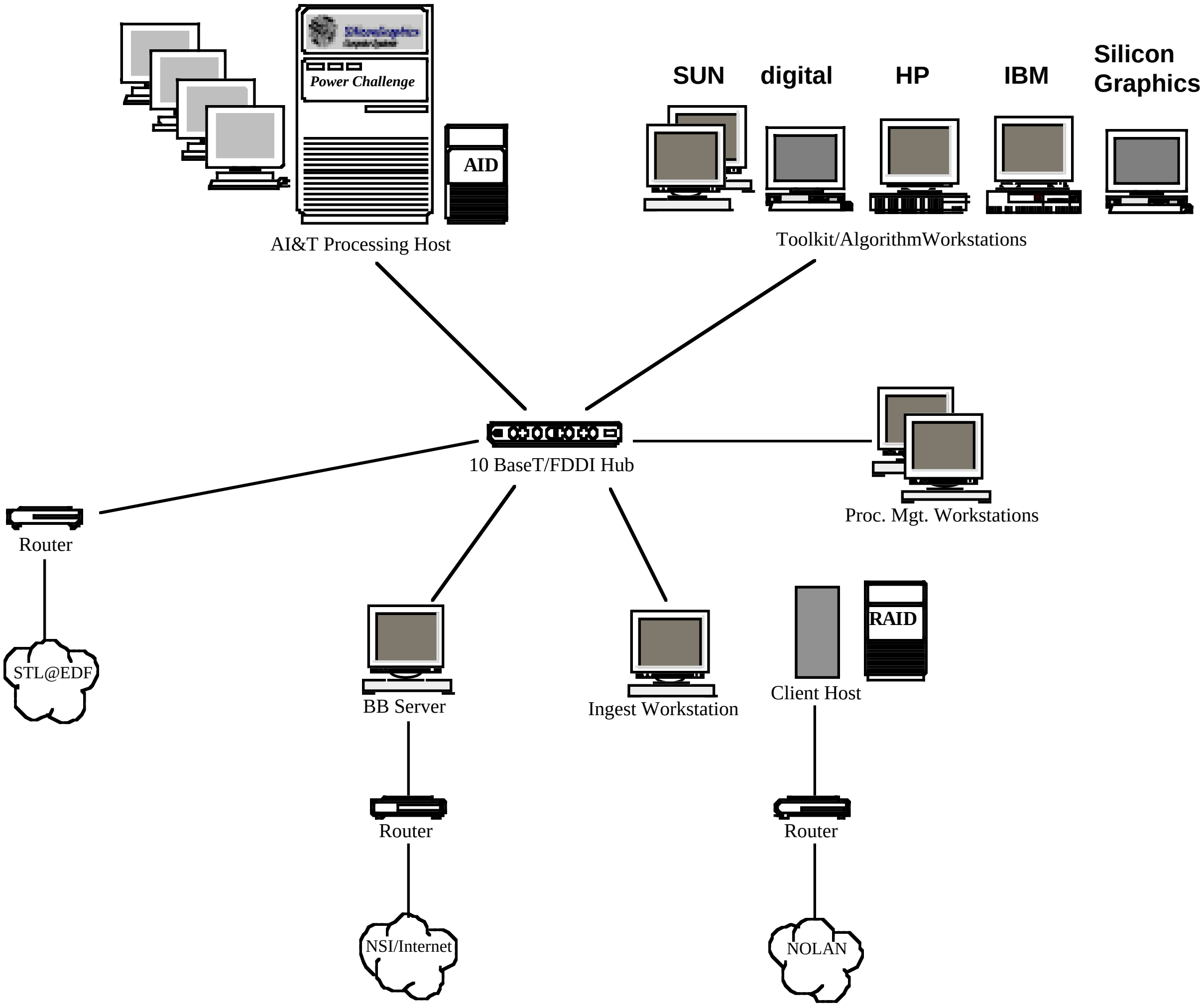

Figure 3-2. IR-1 EDF I&T Hardware Configuration

3.3.4.2 Test Tools and Test Data

SDPS I&T uses test tools for test development, test execution, and test management. Whenever possible test tools from the unit development and unit test environment are used. Additional test tools may be COTS products or developed by the Segment I&T organization. For IR1, SDPS I&T uses the following tools in support of the test environment:

- XRunner and Load Runner–Graphical User Interface (GUI) capture and playback tool used to develop test scripts and facilitate test execution including regression testing
- Requirements Traceability Management (RTM) Tool–This tool provides the means to create, populate, and maintain a data base containing requirements, verification status, and design information. RTM provides the means for SDPS I&T to map Level 4 requirements to SDPS test cases.
- ClearCase Configuration Management Tool–CM tool used to support configuration management for developing, testing and maintaining software.
- Distributed Defect Tracking System (DDTs)–DDTs is the ECS Noncomformance Reporting and Corrective Action (NRCA) system. This discrepancy tracking tool is used to record, track, and report nonconformances encountered during a test session.
- Test management tools –These include tools to record test results and to aid in test result data analysis. These tools include loggers and other recording devices. Some file comparison utility is needed to compare data output with data inputs. A data reduction utility is needed to reduce large amounts of output data, to some meaningful evaluation of the data quality. These test tools for IR1 I&T will be identified in Appendix A of this document.
- Other test specific tools–Specific test tools and test data needed for IR1 SDPS I&T, such as data interface simulators and data generators, are identified in the test cases and Appendix A of this document.

3.3.4.3 Discrepancy Reporting and Resolution

SDPS is obligated to report any noncompliance to Level 4 requirements encountered during SDPS I&T. SDPS I&T will use the DDTs tool (see section 3.3.4.2) for tracking nonconformances. It is the responsibility of the SDPS I&T organization to ensure all testers are trained to use the Noncomformance Reporting and Corrective Action (NRCA) system. The SDPS I&T staff will have the proper authority and access to the NCRA tool before any SDPS I&T activities begin. It is the responsibility of each tester to properly enter all discrepancies encountered during testing into the NRCA system. Once the discrepancy is corrected, regression testing is done to make sure no new problems have been introduced by the fix. If necessary, the tester develops additional tests to ensure the problem is satisfactorily corrected. Quality Assurance (QA) representatives are responsible for audits to ensure reported nonconformances are resolved.

3.3.4.4 Test Items Under Configuration Control

ECS SDPS I&T test documents are controlled but the Data Management Organization (DMO). Software and hardware configurations under test, test data sets, and software and hardware tools used for testing are maintained by CM. SDPS I&T uses the ClearCase tool for configuration management control. It is the responsibility of the CM organization to train all testers to use the

CM tool. The SDPS I&T staff will have the proper authority and access to unit tested components using the CM tool before any SDPS I&T activities begin. Unit-tested components entered in the CM system are accessed by the SDPS testers. These components are verified and integrated by the SDPS I&T staff. Verified segment threads and builds are entered into the CM system upon successful completion of SDPS I&T verification activities. These are made available to the System I&T test team.

If any discrepancies (see section 3.3.4.3) are found during SDPS I&T, CM tracks the product changes and versions that result from correcting discrepancies.

3.4 SDPS I&T Roles and Responsibilities

The SDPS I&T roles include the following test positions and their corresponding responsibilities.

Test Conductor–This will include an SDPS I&T member to conduct test execution. This person is responsible for establishing a sound test configuration before testing takes place. This person is also responsible for collecting test outputs and recording test results. Any problems encountered during testing are entered into the NRCA System by the test conductor.

Test Participants–This will include SDPS I&T members and members of the segment development organization to perform software integration and support test execution. Other supporting organizations include Maintenance and Operations (M&O) and Configuration and Data Management. The ECS maintenance and operations organization will support the test members in the installation and configuration of the test environment and will support the test team if any system faults are encountered during testing. This would include such instances as computer software or hardware failures which cause the test configuration to be corrupted. M&O will be responsible for reconfiguring the system as needed to continue testing. CM will provide a controlled environment for the storing and maintaining of information about the test environment including hardware, software and test tool environments. CM also stores and catalogs test input data and output data.

Test Witnesses–Individuals invited to directly observe test conduct. This includes members from the System I&T organization and the IATO as appropriate in support of System I&T and IATO testing. ESDIS and IV&V personnel are also invited to witness segment test demonstrations.

Test Monitors–The Quality Assurance organization is responsible for reviewing test data, materials, and documentation. These individuals need not be present during test conduct.

3.5 SDPS I&T Release Testing

SDPS I&T verification reviews occur for each ECS formal release. Verification reviews for IR-1 include Test Readiness Reviews (TRRs) and Element Test Reviews (ETRs). Acceptance Testing and Independent Verification and Validation (IV&V) is not performed for IR-1.

TRRs are informal reviews conducted incrementally as portions of the SDPS are unit tested. As software units for each Release are developed and unit tested, informal TRRs are held to determine if the software units are "ready" for integration and test. Test procedures are reviewed at each TRR to determine if they are complete. If the software and test procedures are deemed "ready" the SDPS I&T organization integrates and tests the software.

ETRs are informal reviews conducted incrementally as portions of the SDPS are integrated and tested. At each ETR the results of the portion of the SDPS just integrated and tested are reviewed. The reviews ensure that components are properly integrated and that segment level requirements are met.

A final, formal ETR is held to review the results of all integration and test activities held for the SDPS for that Release. After the final ETR, SDPS software is delivered to System I&T for integration with other segment software.

3.6 SDPS I&T Schedule Overview

The following sections contain SDPS I&T schedules. The emphasis for this document is on IR1 activities. Included is a schedule of SDPS I&T activities for each release and a separate schedule for SDPS I&T activities particular to IR1.

3.6.1 Release Schedule

The following figure (Figure 3-3) shows SDPS I&T activity across all ECS Releases. Only formal reviews are shown. Program releases are indicated in the left most column of the chart. Program milestones are indicated across the top of the chart. For each release, SDPS I&T activities performed for each milestone are indicated. Dates for the milestones can be found in the Level 1 Master Schedule for the ECS Project (194-107-MG1-010).

Release	PDR	IDR	CDR	TRR	ETR	CSR
IR 1	- Produce IR 1 SDPS I&T Plan (DID 319)	n/a	- produce IR 1 SDPS I&T Procedures (draft) (DID 322/DV3)	n/a	- conduct a final ETR for entire segment for IR 1	- IR 1 SDPS I&T Reports (DID 324/DV3)
A	- produce Release A SDPS I&T Plan (DID 319)	n/a	- produce Release A SDPS I&T Procedures (draft) (DID 322/DV3)	- conduct TRR upon completion of all unit development for Release A - produce Release A SDPS I&T Procedures (DID 322/DV3)	- conduct ETR upon completion of each segment level thread/build Turnover to the system test organization for Release A - conduct a final ETR for entire segment for Release A	- Release A SDPS I&T Reports (DID 324/DV3)
B	n/a	- produce Release B SDPS I&T Plan (DID 319)	- produce Release B SDPS I&T Procedures (draft) (DID 322/DV3)	- conduct TRR upon completion of all unit development for Release B - produce Release B SDPS I&T Procedures (DID 322/DV3)	- conduct ETR upon completion of each segment level thread/build Turnover to the system test organization for Release B - conduct a final ETR for entire segment for Release B	- Release B SDPS I&T Reports (DID 324/DV3)
C	n/a	- produce Release C SDPS I&T Plan (DID 319)	- produce Release C SDPS I&T Procedures (draft) (DID 322/DV3)	- conduct TRR upon completion of all unit development for Release C - produce Release C SDPS I&T Procedures (DID 322/DV3)	- conduct ETR upon completion of each segment level thread/build Turnover to the system test organization for Release C - conduct a final ETR for entire segment for Release C	- Release C SDPS I&T Reports (DID 324/DV3)
D	n/a	- produce Release D SDPS I&T Plan (DID 319)	- produce Release D SDPS I&T Procedures (draft) (DID 322/DV3)	- conduct TRR upon completion of all unit development for Release D - produce Release D SDPS I&T Procedures (DID 322/DV3)	- conduct ETR upon completion of each segment level thread/build Turnover to the system test organization for Release D - conduct a final ETR for entire segment for Release D	- Release D SDPS I&T Reports (DID 324/DV3)

Figure 3-3. SDPS I&T Release Schedule

3.6.2 SDPS I&T Schedule for IR1

The following figure (Figure 3-4) shows SDPS I&T activity for IR1.

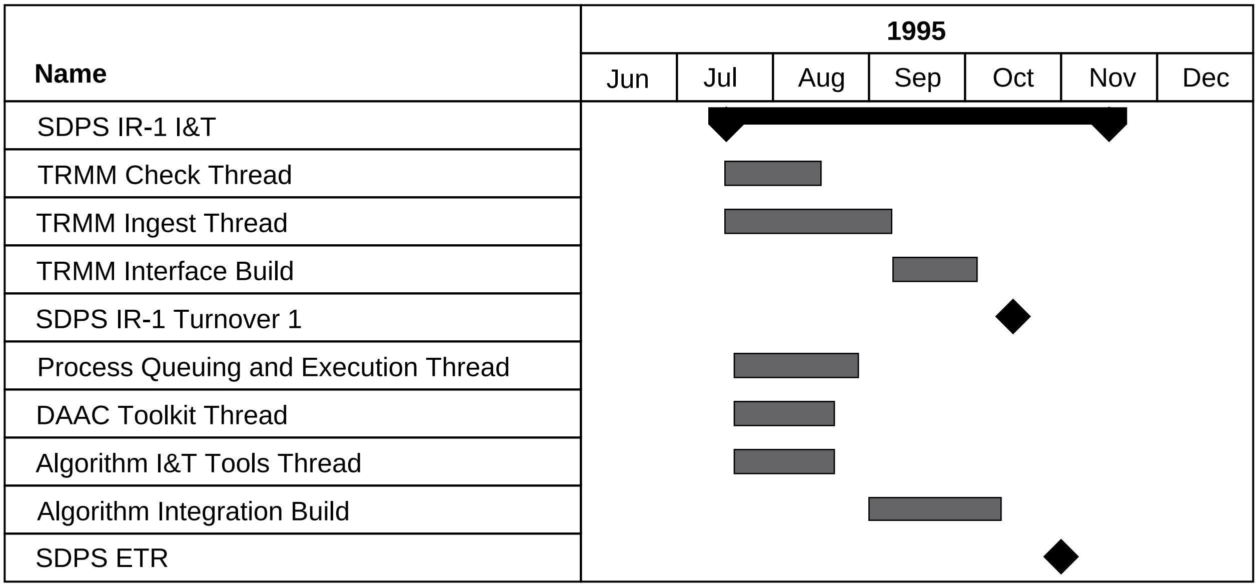

Figure 3-4. SDPS I&T IR1 Schedule

4. SDPS IR1 Test Descriptions

The following sections identify the segment level threads and builds used in Science and Data Processing Segment Integration and Test for Interim Release 1 (shown in Figure 3-1). First, threads are identified. Threads are the aggregation of unit tested components. Each thread demonstrates an SDPS function. Builds are the integration of threads and are identified after each series of threads which make up a build. For each thread and/or build a brief description is given identifying functionality to be tested. Following each description, test cases are identified. These test cases verify segment capabilities allocated to the thread or build. The primary objective of each test case is to demonstrate and evaluate the capabilities of each function as stated in Level 4 requirements.

4.1 Algorithm Integration Tests

The following subsections include the threads and builds identified to support the testing of Algorithm Integration functions. Algorithm Integration testing includes the following threads and builds:

- DAAC Toolkit Thread
- AIT Tools Thread
- Process Queuing and Execution Thread
- Algorithm Integration Build

4.1.1 DAAC Toolkit Thread (TS001)

This thread demonstrates that DAAC Toolkits are successfully used for the science software (algorithm) integration process through swapping of the SCF version of the SDP Toolkit, for the DAAC version of the SDP Toolkit for each science software delivery. The process by which the science software and SDP toolkit integration is verified at AI&T is a three stage process as follows: The SCF version of the SDP Toolkit tools and the science software are first independently compiled, linked and tested for functionality in its native (SCF) environment. Upon delivery of the new science software to the DAAC, the SCF version of the SDP Toolkit is again compiled linked and tested in the DAAC environment with the respective science software delivery. Providing test results are satisfactory, the DAAC version of the SDP Toolkit is then compiled, linked and tested with the same science software at the DAAC, thus assuring a safe and functional environment for the science software delivery. The Science Software to toolkit integration is now considered verified for AI&T. In order to ensure a high degree of portability across a wide variety of platforms the Toolkit conforms to the POSIX standards, enabling science software porting from the development environment (SCF) to a DAAC environment (and through hardware changes as ECS matures).

4.1.1.1 Test Case 1: SCF Toolkit in Native Environment Test (TS001.001)

Test Description

This test case verifies that the SCF version of the SDP Toolkit and the science software can be successfully compiled, linked and tested in its native (SCF) environment and serves as a regression testing of the new/upgraded software. The SCF is the development environment for science software and the originator of all science related changes to a particular algorithm. Upon successful verification of the science software in the SCF environment, the new/upgraded software is available for integration into the DAAC. For this test, an integrated set of test drivers are used to represent the science software.

Test Configuration

Hardware–a suite of SCF representative hardware and operating systems is used to simulate the SCF native environment including:

SUN SparcStation 10 running Sun OS

SGI R3000 running IRIX

HP9000/735 running HP-Ux

IBM RS-600 running AIX

DEC Alpha running OSF1

SUN SparcStation 20/50 running Solaris OS

Software–SCF version of the SDP toolkit functions. They are as follows:

Status Message (SMF) Tools CSC,

Process Control (PC) Tools CSC,

File I/O Tools (Generic File I/O Tools CSC ,

Level 0 Science Data Access Tools and Metadata Tools CSC,

Coordinate System Conversion Tools CSC,

Celestial Body Position (CBP) Tools CSC,

Constant and Unit Conversion (CUC) Tools CSC,

Geo-Coordinate Transformation (GCT) Tools CSC,

Ancillary Data Tools CSC,

Memory Management (MEM) Tools CSC,

and Time and Date Conversion (TD) Tools CSC.

Test Support Required

Compilers (FORTRAN 77, FORTRAN 90, and C), File Comparison Tools, Toolkit Test Drivers

Test Input

Inputs to this test include calls to SDP Toolkit functions. Parameters for these calls are contained in input files for each test driver.

Expected Test Outputs

Outputs to this test include: compilation status messages and logs, binary code, and toolkit return messages and data recorded in the test driver output files.

Success Criteria

Using the SCF version of the SDP Toolkit, each science software delivery developed at the SCF should compile, link, and produce the expected test results in the SCF environment.

4.1.1.2 Test Case 2: SCF Toolkit in DAAC Environment Test (TS001.002)

Test Description

This test case verifies that the SCF version of the SDP Toolkit and the science software can be successfully compiled, linked and tested in the DAAC environment. Upon successful verification of the science software in the SCF environment, the new/upgraded software is made available for integration into the DAAC. The DAAC receiving the science software upgrade, compiles, links and tests the science software delivery with the SCF version of the SDP Toolkit in the DAAC configuration. Test results are examined and compared with the expected test results delivered in the software delivery package. For this test, an integrated set of test drivers are used to represent the science software and the input and output files from Test Case 1 represents the test data delivered in the software delivery package.

Test Configuration

Hardware–Representative DAAC Processing host (Symmetrical multiprocessor or cluster workstation)

Software–SCF version of the SDP toolkit functions. They are as follows:

Status Message (SMF) Tools CSC,

Process Control (PC) Tools CSC,

File I/O Tools (Generic File I/O Tools CSC,

Level 0 Science Data Access Tools and Metadata Tools CSC,

Coordinate System Conversion Tools CSC,

Celestial Body Position (CBP) Tools CSC,

Constant and Unit Conversion (CUC) Tools CSC,

Geo-Coordinate Transformation (GCT) Tools CSC,

Ancillary Data Tools CSC,

Memory Management (MEM) Tools CSC,

and Time and Date Conversion (TD) Tools CSC.

Data–Input and output files from Test Case 1

Test Support Required

Compilers (FORTRAN 77, FORTRAN 90, and C), File Comparison Tools, Toolkit Test Drivers

Test Input

Inputs to this test include calls to SDP Toolkit functions. Parameters for these calls are contained in input files for each test driver.

Expected Test Outputs

Outputs to this test include: compilation status messages and logs, binary code, and toolkit return messages and data recorded in the test driver output files.

Success Criteria

Using the SCF version of the SDP Toolkit, each science software delivery developed at the SCF should compile, link, and produce the expected test results in the DAAC environment. Outputs from Test Case 1 are compared with Test Case 2 outputs to ensure the SCF version of the SDP toolkit provides consistent results in the SCF and DAAC environment.

4.1.1.3 Test Case 3: Test DAAC Toolkit in DAAC Environment (TS001.003)

Test Description

This test case verifies that the DAAC version of the SDP Toolkit and the science software can be successfully compiled, linked and tested in the DAAC environment. Upon successful verification of the science software in the DAAC environment using the SCF version of the SDP Toolkit, the new/upgraded software is recompiled, linked and tested at the DAAC again, using the DAAC version of the SDP Toolkit. Test results are examined and compared with the expected test results delivered in the software delivery package. For this test, an integrated set of test drivers are used to represent the science software and the input and output files from Test Case 1 represents the test data delivered in the software delivery package.

Hardware–Representative DAAC Processing host (Symmetrical multiprocessor or cluster workstation)

Software–DAAC version of the SDP toolkit functions. They are as follows:

Status Message (SMF) Tools CSC,

Process Control (PC) Tools CSC,

File I/O Tools (Generic File I/O Tools CSC ,

Level 0 Science Data Access Tools and Metadata Tools CSC,

Coordinate System Conversion Tools CSC,

Celestial Body Position (CBP) Tools CSC,

Constant and Unit Conversion (CUC) Tools CSC, Geo-Coordinate Transformation (GCT) Tools CSC,

Ancillary Data Tools CSC,

Memory Management (MEM) Tools CSC

and Time and Date Conversion (TD) Tools CSC.

Data–Input and output files from Test Case 1

Test Support Required

Compilers (FORTRAN 77, FORTRAN 90, and C), File Comparison Tools, Toolkit Test Drivers

Test Input

Inputs to this test include calls to SDP Toolkit functions. Parameters for these calls are contained in input files for each test driver.

Expected Test Outputs

Outputs to this test include: compilation status messages and logs, binary code, and toolkit return messages and data recorded in the test driver output files.

Success Criteria

Using the DAAC version of the SDP Toolkit, each science software delivery developed at the SCF should compile, link, and produce the expected test results in the DAAC environment. Outputs from Test Case 1 are compared with Test Case 3 outputs to ensure the DAAC version of the SDP toolkit provides results consistent with the SCF version.

4.1.2 AIT Tools Thread (TS002)

This thread demonstrates that AIT Tools successfully support Version 1 Algorithm I & T. Science software developed at the SCF, is delivered to a DAAC. It is the responsibility of the DAAC to accept the delivery of science software, and to evaluate the science software for compatibility with the DAAC environment. To support Algorithm I&T in evaluation of science software, tools provide the following functions: documentation viewing, standards checking, code execution using make files, results comparison, configuration management, report generation, and display of product metadata and on-line documentation. They also include POSIX OS utilities, Internet utilities, and data staging tools.

4.1.2.1 Test Case 1:Science Software (Algorithm) Delivery Verification Test (TS002.001)

Test Case Description

This test case demonstrates the ability to receive a science software delivery package from the SCF electronically. In IR1 the Ingest capabilities are not required to be in place. Therefore the receipt of deliveries shall occur via the Internet for this test.

Test Configuration

Hardware–workstation

Software–CSS Services (FTP, TCP/IP), Report Generation Tools CSC

Data–Science software deliveries

Test Support Required

N/A

Test Input

The inputs to this test include valid and invalid Internet address specifications of science software delivery packages (in compressed tar format), and delivery notifications.

Expected Test Output

The outputs to this test include: status logs, receipt of delivery packages, and receipt acknowledgment.

Success Criteria

For each science software delivery notification received from the SCF, a successfully transmitted delivery package shall be received with all its advertised components and a receipt acknowledgment shall be produced.

4.1.2.2 Test Case 2: Viewing Science Software Documentation Test (TS002.002)

Test Case Description

This test case demonstrates the capability to display and print documentation included in a science software delivery in PostScript, Tex, ASCII, HTML and RTF formats.

Test Configuration

Hardware–workstation, printer, local disk storage

Software–Documentation Viewing Tools CSC

Data–PostScript, Tex, ASCII, HTML and RTF files

Test Support Required

N/A

Test Input

Input to this test is valid and invalid documentation files of various text formats (specifically PostScript, TeX, ASCII, HTML and RTF).

Expected Test Outputs

Outputs to this test include: legible documentation display file or printout, and status messages.

Success Criteria

For any documentation received in PostScritpt, TeX, ASCII, HTML or RTF formats, a legible display or printout shall be produced.

4.1.2.3 Test Case 3: Code Standard Checking Test - FORTRAN 77 Code (TS002.003)

Test Case Description

This test case demonstrates the capability to:

(1) verify that software and scripts included in a science software delivery and written in FORTRAN 77 code, are in compliance with ANSI, POSIX and ESDIS standard specifications. The following exceptions apply:

INCLUDE statement, BYTE data type, DO WHILE, EXIT, ENDO, STRUCTURE data types, names up to 31 characters long, IMPLICIT NONE statement, block IF with ELSE IF and END IF, in-line comments, extended character set which includes lower case, underscore, left and right angle brackets, quotation mark, percent sign, and ampersand, initialization of data in declaration, and long line extensions beyond 72 characters per line.

(2) verify that the first line of each science software script specifies either the "C", Bourne, Korn, Perl, or POSIX shell.

(3) verify that all science software code contains headers and follows the coding standards established in the "Data Production Software and SCF Standards and Guidelines" document.

(4) verify the capability to generate report files of standards checking.

Test Configuration

Hardware–workstation, local disk storage

Software–Standards Checkers CSC, Report Generation Tool CSC

Data–Science Software code

Test Support Required

N/A

Test Input

Inputs to this test include: FORTRAN 77 software and scripts included in a science software delivery.

Expected Test Outputs

Outputs to this test include: compliance reports, and status messages

Success Criteria

For each science software delivery containing FORTRAN 77 code, a report shall be generated that logs the adherence/non-adherence to POSIX, ANSI and ESDIS standards.

4.1.2.4 Test Case 4: Code Standard Checking Test - FORTRAN 90 Code (TS002.004)

Test Case Description

This test case demonstrates the capability to:

(1) verify that software and scripts included in a science software delivery are in compliance with ANSI, POSIX and ESDIS standard specifications for FORTRAN 90 code. The following exception applies: usage of DO WHILE.

(2) verify that the first line of each science software script specifies either the "C", Bourne, Korn, Perl, or POSIX shell.

(3) verify that all science software code contains headers and follows the coding standards established in the "Data Production Software and SCF Standards and Guidelines" document.

(4) verify that the capability to generate report files of standards checking.

Test Configuration

Hardware–workstation, local disk storage

Software–Standards Checkers CSC, Report Generation Tools CSC

Data–Science Software code

Test Support Required

N/A

Test Input

Inputs to this test include: FORTRAN 90 software and scripts included in a science software delivery.

Expected Test Outputs

Outputs to this test include: compliance reports, and status messages

Success Criteria

For each science software delivery containing FORTRAN 90 code, a report shall be generated that logs the adherence/non-adherence to POSIX, ANSI and ESDIS standards.

4.1.2.5 Test Case 5: Code Standard Checking Test - "C" Code (TS002.005)

Test Case Description

This test case demonstrates the capability to:

(1) verify that software and scripts included in a science software delivery and written in "C" code, are in compliance with ANSI, POSIX and ESDIS standard specifications.

(2) verify that the first line of each science software script specifies either the "C", Bourne, Korn, Perl, or POSIX shell.

(3) verify that all science software code contains headers and follows the coding standards established in the "Data Production Software and SCF Standards and Guidelines" document.

(4) verify the capability to generate report files of standards checking.

Test Configuration

Hardware–workstation, local disk storage

Software–Standards Checkers CSC, Report Generation Tools CSC

Data–Science Software code

Test Support Required

N/A

Test Input

Inputs to this test are compliant and non-compliant "C" code and/or scripts included in a science software delivery.

Expected test Outputs

Outputs to this test include: compliance reports, and status messages

Success Criteria

For each science software delivery containing "C" code, a report shall be generated that logs the adherence/non-adherence to POSIX, ANSI, and ESDIS standards.

4.1.2.6 Test Case 6: Code Standard Checking Test - Ada Code (TS002.006)

Test Case Description

This test case demonstrates the capability to:

(1) verify that software and scripts included in a science software delivery and written in Ada code, are in compliance with ANSI, POSIX and ESDIS standard specifications.

(2) verify that the first line of each science software script specifies either the "C", Bourne, Korn, Perl, or POSIX shell.

(3) verify that all science software code contains headers and follows the coding standards established in the "Data Production Software and SCF Standards and Guidelines" document.

(4) verify the capability to generate report files of standards checking.

Test Configuration

Hardware–workstation, local disk storage

Software–Standards Checkers CSC, Report Generation Tools CSC

Data–Science Software code

Test Support Required

N/A

Test Input

Inputs to this test are compliant and non-compliant Ada code and/or scripts included in a science software delivery.

Expected test Outputs

Outputs to this test include: compliance reports, and status messages

Success Criteria

For each science software delivery containing Ada code, a report shall be generated that logs the adherence/non-adherence to POSIX, ANSI, and ESDIS standards.

4.1.2.7 Test Case 7: Code Standard Checking Test - SDP Toolkit Usage Requirements (TS002.007)

Test Case Description

This test case demonstrates the capability to:

(1) verify that SDP toolkit usage requirements are adhered to.

- the PGE must set the required environment variables
- the Process Control File must exist
- the toolkit functions PGS_SMF_SendRuntimeData, PGS_SMF_Send Status Report, PGS_MEM_Create are not called more than once per PGE
- the PGE must reinitialize the pointers to null for memory blocks freed with the PGS_MEM_Free or PGS_MEM_FreeAll SDP Toolkit functions before reuse in calls to PGS_MEM_Malloc, PGS_MEM_Calloc, or PGS_MEM_Realloc

- the first shared memory tool (PGS_MEM_Shm*) used must be the function PGS_MEM_ShmCreate

(2) verify that the first line of each science software script specifies either the "C", Bourne, Korn, Perl, or POSIX shell.

(3) verify that all science software code contains headers and follows the coding standards established in the "Data Production Software and SCF Standards and Guidelines" document.

(4) verify the capability to generate report files of standards checking.

Test Configuration

Hardware–workstation, local disk storage

Software–Standards Checkers CSC, Report Generation Tools CSC

Data–Science Software code

Test Support Required

N/A

Test Input

Inputs to this test include: standard and non-standard science software deliveries

Expected Test Outputs

Outputs to this test include: status messages, logs and analyzed science software code.

Success Criteria

Science software code should pass all standards checking without errors.

4.1.2.8 Test Case 8: Code Standard Checking - Extract Required Header Data Test (TS002.008)

Test Case Description

This test case demonstrates the ability to extract required header data from science software source code.

Test Configuration

Hardware–workstation, local storage device

Software–Standards Checkers CSC, Report Generation Tools CSC

Data–Science Software code

Test Support Required

N/A

Test Input

Inputs to this test include: standard and non-standard science software deliveries

Expected Test Outputs

Outputs to this test include status messages and logs of analyzed science software header information.

Success Criteria

For all Science software code submitted, required header information shall be extracted and presented in a standards checking results report.

4.1.2.9 Test Case : Development Environment Test (TS002.009)

Test Case Description

This test case demonstrates that the AIT environment has the capability to compile and link object code into binary executables through the use of POSIX.2 software development utilities such as "make", with code written in "C", FORTRAN 77, FORTRAN 90 and Ada. The ability to edit, use an interactive debugger, use on-line documentation/printed documentation, and screen capture capability in support of these languages shall be demonstrated. In support of "C" language development, the utilities lex and yacc are available for use.

Test Configuration

Hardware–workstation, local disk storage

Software–AITHW CI, Report Generation CSC

Data–science software code

Test Support Required

N/A

Test Input

Test inputs include compilable and non-compilable software in "C", FORTRAN77, FORTRAN90 and Ada.

Expected Test Outputs

Test outputs include compilation logs, status messages and logs.

Success Criteria

For science software code in written in "C", FORTRAN77, FORTRAN90 and Ada, a binary executable shall be produced. Capabilities to edit, use debugger, use documentation, use screen capture capability, and use lex and yacc shall be proven.

4.1.2.10 Test Case 10: File Comparison Test (TS002.010)

Test Case Description

This test case demonstrates the ability to read ASCII, binary, HDF or user defined custom data format and find all differences existing between any two files within a specified absolute or relative threshold and generate reports of the file comparison results.

Test Configuration

Hardware–workstation, local disk storage,

Software–File Comparison Utility CSC, Report Generation Tools CSC

Data–science software code

Test Support Required

N/A

Test Input

The inputs to this test include ASCII, binary, HDF and user defined custom data formats.

Expected Test Outputs

The outputs to this test include status logs/reports and files of data differences within a specified tolerance.

Success Criteria

Given two data files and a specified tolerance, existing differences shall be displayed in status, log, file or report format.

4.1.2.11 Test Case 11: Configuration Management Test (TS002.011)

Test Case Description

This test case demonstrates the ability to configure deliveries to the AIT environment and track problems that are identified in the configured environment.

Test Configuration

Hardware–workstation, local disk storage

Software–AITHW CI, Report Generation Tools CSCl

Data–science software code

Test Support Required

N/A

Test Input

Inputs to this test are science software deliveries.

Expected Test Outputs

Outputs to this test are configured science software deliveries, status logs and messages. In the case of errors Problem Tracking logs and reports will be reproduced.

Success Criteria

This test is deemed successful if all configurable components of a science software delivery are configured.

4.1.2.12 Test Case 12: Status Tracking and Report Generation Test (TS002.012)

Test Case Description

This test case demonstrates the ability to track the status of the science software integration process, and to demonstrate the ability of authorized users to examine the produced logs and reports at any time from receipt through the end of acceptance. The status tracking processing shall entail the usage of plotting tools, spreadsheets, drawing and word processors to produce logs, reports and status information (according to specified templates) throughout the integration phase.

Test Configuration

Hardware–workstation, local disk storage

Software–Report Generation Tools CSC

Data–text files, logs

Test Support Required

N/A

Test Input

Inputs to this test should include status tracking logs and spreadsheets.

Expected Test Outputs

Outputs to this test should include new and updated status tracking logs and spreadsheets.

Success Criteria

This test is successful if updated status information can be produced of AIT integration.

4.1.2.13 Test Case 13: Operating System and Utilities Test (TS002.013)

Test Case Description

This test verifies that the operating system for each UNIX platform in the AI&T and development environment:

- is in compliance with POSIX.2 standards
- has the "C", Bourne, and Korn shells installed.
- has the GNU utilities perl, emacs, gzip, tar installed
- has the POSIX.2 user portability utilities man, vi
- has on-line documentation (when available) or printed documentation for all tools installed

Test Configuration

Hardware–symmetrical multiprocessor or cluster of workstations (most likely SGI Powerchallenge, DEC, SunStation or IBM SP2)

Software–AITHW CI, Standards Checkers CSC

Data–N/A

Test Support Required

N/A

Test Input

The input to this test includes multiple UNIX platform operating systems.

Expected Test Outputs

The output to this test shall include status logs and messages.

Success Criteria

All operating systems that are tested shall show evidence of POSIX.2 compliance, the existence of the "C", Bourne, and Korn shells, the existence of gnu utilities, portability utilities and documentation for all tools.

4.1.2.14 Test Case 14: Internet Utilities Test (TS002.014)

Test Case Description

This test verifies the capability to browse WWW, to read from and post USENET newsgroups, read and send electronic mail via the Internet and provide access to a gopher and WAIS clients.

Test Configuration

Hardware–workstation

Software–CSS Services (WWW, WAIS, USENET, MOSIAC, telnet etc.)

Data–N/A

Test Support Required

N/A

Test Input

Inputs to this test include URL locations and email messages.

Expected Test Outputs

Outputs to this test include Email messages and URL locations.

Success Criteria

This test is successful if the ability to receive and distribute news, email, and WWW URLs over the Internet is demonstrated.

4.1.2.15 Test Case 15: Product Metadata Test (TS002.015)

Test Case Description

This test case verifies the capability to view metadata associated with a data file and write it to a report file.

Test Configuration

Hardware–workstation, terminal, magnetic storage device

Software–Product Metadata Display Tool CSC, Report Generation Tools CSC

Data–product metadata

Test Support Required

N/A

Test Input

Inputs to this test include data file metadata.

Expected Test Outputs

The output to this test includes legible metadata file displays and report files.

Success Criteria

All data file metadata shall be able to produce a legible report file via the use of the SDP toolkit.

4.1.2.16 Test Case 7: Profiling Test (TS002.016)

Test Case Description

This test demonstrates the ability of the performance analysis and resource management tools to measure, record and report the following performance statistics of a process: CPU time, CPU time of each procedure, memory usage, disk space usage, number of I/O accesses to each of its input or output data files, wall clock time of a process, wall clock time of a procedure, and number of page faults. These statistics can be printed or saved as soft copy.

Test Configuration

Hardware –workstation, terminal,

Software–Profiling Tools CSC

Data–Science Software Delivery software (shell scripts, executables)

Test Support Required

N/A

Test Input

AI&T performance metadata, AIT resource metadata. A series of tests are performed using data which produces varying results (CPU time CPU time of each procedure, memory usage, disk space usage, and the number of I/O accesses to each input or output data files).

Expected Test Outputs

Report which chronicles the performance statistics of a designated process which can be displayed to console, printed to hard copy or saved as soft copy.

Success Criteria

For all performance/resource metadata regarding a specific process submitted, a report will be produced of performance statistics. Statistics will be verified for accuracy.

4.1.3 Queuing and Execution Thread (TS003)

This thread demonstrates the ability to accept, queue, process and monitor the processing requests. It supports the basic graphical user interface (GUI) to facilitate manual submittal of data processing requests. Other capabilities include queuing, PGE execution, and monitoring process status.

4.1.3.1 Test Case 1: Manual Submittal of Data Processing Requests (TS003.001)

Test Case Description

This test verifies the capability to facilitate manual submittal of Data Processing Requests via a GUI. The request is submitted to a template processing script which is set up for each Product

Generation Executive (PGE). The information contained in the request is substituted into the processing script encapsulating the PGEs. The processing script contains Process Control Information accessed by SDP Toolkit. Resource request and staging directive can also be included in the script.

Test Configuration

Hardware–Processing Queue Control workstation

Software–Queuing and Execution Tool (Processing Management, Processing Queue Management)

Data– simulated CERES and LIS L0 data granule, simulated PGE

Test Support Required

Queuing and Execution Tool uses CHAINS interface and Distributed Queuing System (DQS). CHAINS software controls and monitors algorithm execution. DQS Master maintains the job-list and monitors the queue. They are installed onto a Processing Queue Control workstation which has MOTIF graphics libraries.

Test Inputs

A Data Processing Request is entered manually via the GUI.

Expected Test Outputs

Processing script is created.

Success Criteria

The processing script is created based on the information of the entered Data Processing Request. Default variables are used if no information is provided in the Data Processing Request.

4.1.3.2 Test Case 2: Process Queue (TS003.002)

Test Case Description

This test verifies the capability to accept and distribute processes to the relevant processing resources based on the specification of the resource requirements and load balancing. One or more queues may be maintained on each of the processing resources. Queue can be configured to allow concurrent PGE execution.

Test Configuration

Hardware–Processing Queue Control workstations, representative DAAC Processing hosts (Symmetrical multiprocessors or cluster workstations)

Software–Queuing and Execution Tool (Processing Management, Processing Queue Management)

Data–simulated CERES and LIS L0 data granule, simulated PGE

Test Support Required

DQS Master and CHAINS interface are installed onto the machine which has MOTIF graphics libraries and is used for processing queue control and management. One or more workstations are set up as processing resources with DQS Exec installed. They are responsible for PGE execution.

Test Inputs

Processing scripts are submitted via GUI.

Expected Test Outputs

Data processing requests are dispatched from the job-list to the queues on the processing resources.

Success Criteria

Operator GUI display shows a list of PGEs in waiting or queued for execution.

4.1.3.3 Test Case 3: PGE Execution (TS003.003)

Test Case Description

This test verifies the capability to execute the PGE execution when resource is available.

Test Configuration

Hardware–Processing Queue Control workstation, representative DAAC Processing hosts (Symmetrical multiprocessors or cluster workstations)

Software–Queuing and Execution Tool (Processing Management, Processing Queue Management)

Data–simulated CERES and LIS L0 data granule, simulated PGE

Test Support Required

DQS Master and CHAINS interface are installed onto the machine which has MOTIF graphics libraries. One or more representative DAAC Processing hosts are set up as processing resources with DQS Exec installed.

Test Inputs

Processing script is submitted via GUI.

Expected Test Outputs

Data processing request is dispatched from the job-list to the queue on the processing resources. When resource is available, execution is initiated.

Success Criteria

PGE is executing. Operator GUI display shows PGE is in executing state.

4.1.3.4 Test Case 4: Status of Data Processing Request (TS003.004)

Test Case Description

This test verifies the capability to provide status of data processing request by continuously monitor the queue and processing resources. The basic information are: unstarted, queue, executing, complete, and error . They are displayed on the operator GUI.

Test Configuration

Hardware–Processing Queue Control workstation, representative DAAC Processing hosts (Symmetrical multiprocessors or cluster workstations)

Software–Queuing and Execution Tool (Processing Management, Processing Queue Management)

Data–simulated CERES and LIS L0 data granule, simulated PGE

Test Support Required

DQS Master and CHAINS interface are installed onto the machine which has MOTIF graphics libraries. One or more representative DAAC Processing hosts are set up as processing resources with DQS Exec installed.

Test Inputs

PGE chain is displayed on Operator machine. One data processing request is allowed to successfully complete. Then cancellation requests are applied onto other data processing requests that are in queued and executing state.

Expected Test Outputs

Operator GUI display of PGE chain shows the processing status. The status is dynamically updated.

Success Criteria

GUI display shows status of the data processing requests. The data processing requests are removed as the result of the cancellation requests. The one that is allowed to complete should have its status updated as complete.

4.1.3.5 Test Case 5: Processing System Initialization and Shutdown Test (TS003.005)

Test Case Description

This test demonstrates the ability to initialize and shutdown the Processing system in an orderly fashion. Using a system account with appropriate privileges, commands are entered to initialize the Processing system. The system status is monitored to determine the status of the initialization. Once the system is successfully initialized, commands are entered for orderly shutdown. Again, the system is monitored to determine shutdown status.

Test Configuration

Hardware–Processing Queue Control workstations, representative DAAC Processing hosts (Symmetrical multiprocessors or cluster workstations)

Software–Queuing and Execution Tool (Processing Management, Processing Queue Management)

Data–N/A

Test Support Required

N/A

Test Inputs

A series of commands are entered for initialization, shutdown and system monitoring.

Test Expected Outputs

System monitoring to confirm the state of the Processing system.

Success Criteria

The system commands are accepted and successful initialization and shutdown of the system is confirmed by system monitoring.

4.1.3.6 Test Case 6: Science Processing Operating System and Utilities Test (TS003.006)

Test Case Description

This test case verifies that the operating system for each UNIX platform in the science processing environment is in compliance with POSIX.2 standards and has C shell, Korn shell, and Bourne shell installed by running C shell, Korn shell and Bourne shell through the POSIX checker and invoking each shell. Utilities such as: perl, emacs, gzip, tar, imake, prof, gprof, nm, man, vi, make, lex and yacc are also available.

Test Configuration

Hardware–Processing Queue Control workstations, representative DAAC Processing hosts (Symmetrical multiprocessors or cluster workstations)

Software–Queuing and Execution Tool (Processing Management, Processing Queue Management)

Data–N/A

Test Support Required

POSIX standards checker

Test Input

UNIX commands to invoke C shell, Korn shell, Bourne shell, perl, emacs, gzip, tar, imake, prof, gprof, nm, man, vi, make, lex and yacc.

Expected Test Outputs

Active POSIX.2 compliant C shell, Korn shell, and Bourne shell; status logs and messages

Success Criteria

For each representation of a Processing hardware/software environment, evidence of POSIX.2 compliance shall be shown and a functioning UNIX shells and utilities will be proven through logs and messages generated by the POSIX checker and status messages generated by invoking each command.

4.1.3.7 Test Case 7: Science Processing Development Environment Test (TS003.007)

Test Case Description

This test case demonstrates that Science Processing environment has the capability to compile and link object code written in C, FORTRAN 77, and ADA into executables. Other installed tools are dynamic analyzer for memory leaks and interactive source level debugger.

Test Configuration

Hardware–Processing Queue Control workstations, representative DAAC Processing hosts (Symmetrical multiprocessors or cluster workstations)

Software–Queuing and Execution Tool (Processing Management, Processing Queue Management)

Data–N/A

Test Support Required

Source code contains extensions specified in the Data Production S/W and SCF Standards and Guidelines

Test Input

Source codes are compiled and linked; debugger is turned on and memory leak analyzer is used while running the executables.

Expected Test Outputs

Binary executables, debugger report is displayed on the screen, number of bytes of memory leaked is reported if there are any.

Success Criteria

For any source code submitted, a successful compile and link shall occur and, as a result, binary executable will be generated. Ability to use the debugger and memory leak analyzer while running this executable is demonstrated.

4.1.3.8 Test Case 8: Science Processing Documentation Test (TS003.008)

Test Case Description

This test case demonstrates that Processing environment has either on-line display or hard copy print of documentation for each tool installed. These shall be displayed on console or in hard copy print.

Test Configuration

Hardware–Processing Queue Control workstations, representative DAAC Processing hosts (Symmetrical multiprocessors or cluster workstations)

Software–Queuing and Execution Tool (Processing Management, Processing Queue Management)

Data–N/A

Test Support Required

N/A

Test Input

Calls to invoke all on-line documentation for utilities and tools in Processing environment

Expected Test Outputs

On-line display or hard copy print of documentation of each installed tool.

Success Criteria

For each tool installed, the existence of associated on-line and/or printed documentation will be displayed and verified by inspection.

4.1.4 Algorithm Integration Build (BS001)

The Algorithm Integration Build integrates the functionality of the DAAC Toolkit Thread, the AIT Tools Thread, and the Queuing and Execution Thread. This build demonstrates the capability to perform Algorithm I & T using DAAC Toolkits and AIT Tools to verify that the science software can operate successfully, efficiently and safely in the DAAC environment. It should be noted that quality of the product or the science of the algorithm is not verified here, only the verification that the algorithm operates safely in the DAAC environment. Product quality and quality of the science of the algorithm is the responsibility of the SCF.

4.1.4.1 Test Case 1: Ingestion of Delivery Package (BS001.001)

Test Case Description

This test case verifies that a Science Software delivery (Algorithm) package can be retrieved electronically from the Internet. Notification of delivery is accomplished by the receipt of an email notification containing the FTP address of the delivery package in the transfer directory. For this test, representations of delivery notifications, in a transfer directory, containing FTP addresses of Science Software delivery package locations are specified. The science software delivery packages are retrieved electronically via the Internet. A science software delivery package is retrieved in the form of a compressed tar file and should contain a delivery memo, documentation, software delivery, scripts, test plans, test data and expected test results. Upon successful receipt, the science software delivery is placed in the specified delivery directory and is untarred. Successful receipt of the delivery package is recorded, email notification of receipt status is relayed and status log files are updated. The delivery package is validate, integrated and tested in the simulated DAAC environment.

Test Configuration

Hardware–workstation, terminal, magnetic storage device

Software–FTP, TCP/IP, email, UNIX utilities

Data– science software (algorithm) delivery package

Test Support Required

Internet Utilities, report generation tools (word processor, spreadsheets, plotting programs, drawing tools)

Test Input

Inputs to this test are email Science software delivery notifications from the SCF containing valid and invalid ftp addresses of delivery packages in the transfer directory and other locations.

Expected Test Outputs

Outputs to this test include fully transmitted delivery packages in the form of an uncompressed tar file, email acknowledgment of receipt message, status log files.

Success Criteria

For each science software delivery notification from the SCF, a delivery package is retrieved (compressed tar format), stored in the delivery directory at the DAAC and is uncompressed and untarred. The status log file is updated to reflect that the advertised delivery has been successfully received and untarred. Confirmation of receipt is successfully sent to the SCF.

4.1.4.2 Test Case 2: Evaluation, Inspection and Verification of the Science Software Delivery (BS001.002)

Test Case Description

This test case verifies that a Science Software (algorithm) delivery package can be inspected for completeness (as specified in the delivery memo), and that the science software is compatible in the ECS environment. Verification of compatibility in the ECS environment encompasses proving that integration of the algorithm with currently running operational DAAC software interfaces can be performed, that preliminary performance statistics can be gathered, and that the mechanics to check for algorithm reliability and operational safety can be performed satisfactorily. Functionality that is confirmed in this test is the ability to display and print documentation provided in the delivery, to verify that all code deliveries are compliant with POSIX, ANSI and ESDIS standards, that compilation capabilities exist, and that status logging can be performed throughout the entire process. For this test, representative science software deliveries in uncompressed file format are accepted, resulting in a configured and operationally acceptable science software delivery and an associated updated status logs.

Test Configuration

Hardware– Terminal, workstation, printer, magnetic storage device

Software–Static analyzer, lint detectors, custom ECS software (standards checkers, file comparison tool), Perl scripts.

Data–Science Software Delivery

Test Support Required

Word processors, compilers and linkers, UNIX tools, report generation tools (spreadsheets, plotting programs, drawing tools), configuration management (Clearcase)

Test Input

Inputs to this test include partial and complete science software deliveries in compressed tar format, deliveries that contain compliant and noncompliant software according to POSIX, ANSI and ESDIS standards.

Expected Test Outputs

Outputs to this test include configured, uncompressed, untarred science software delivery packages , supporting documentation in hard copy and on-line formats, and status logs.

Success Criteria

For each science software delivery package, all components listed in the delivery memo are accounted for, all software is POSIX, ANSI and ESDIS compliant (per the standards checkers), is successfully configured and status logs are produced of each transition.

4.1.4.3 Test Case 3: Science Software Delivery Integration Test (BS001.003)

Test Case Description

This test case verifies that the portability of the science software (algorithm) delivery from the SCF environment to its perspective DAAC environment is achieved satisfactorily, that the system-level interfaces and value added features supplied by the SDP toolkit are functional in the DAAC environment through swapping of the SCF toolkit for the DAAC toolkit. The science software, delivery is compiled and linked with the SCF version of the SDP Toolkit and the test plans provided in the delivery are executed. Results from this test are compared to the expected results supplied in the delivery. Upon successful completion of this phase, the science software delivery is compiled and linked with the DAAC version of the SDP Toolkit. Results from this test are also compared to the expected results supplied in the delivery. Throughout this process, preliminary performance statistics are gathered and status logs are produced. For this test representative science software deliveries will be used to verify the swapping of the SCF toolkit for the DAAC toolkit.

Test Configuration

SCF Toolkit Hardware Configuration–SUN SparcStation 10, SGI, HP, IBM, DEC,SUN SparcStation 20/50

DAAC Toolkit Hardware Configuration –symmetrical multiprocessor or workstation cluster

Software –C, FORTRAN77, FORTRAN90

Data– SDP toolkit tools, Science Software delivery

Test Support Required

Compilers, linkers, file comparison tools, report generation tools (word processors, spreadsheets, drawing tools), configuration management (ClearCase).

Test Input

Inputs to this test include compilable and non-compilable science software algorithms in the SCF toolkit environment and the DAAC toolkit environment.

Expected Test Outputs

Output to this test include a completely verified environment for the operational support of the science software at the DAAC, preliminary performance statistics, and status logs.

Success Criteria

For each science software delivery, swapping of the SCF toolkit for the DAAC toolkit is done successfully, test plans supplied with the delivery are executed and expected results are received, preliminary performance statistics are gathered and status logs are produced.

4.1.4.4 Test Case 4: Acceptance Testing of the Science Software Delivery (BS001.004)

Test Case Description

This test case verifies that the commissioning period of the science software is satisfied, the science software runs safely within the Processing Subsystem, the product outputs produced in the SCF environment can also be produced in the DAAC environment and that the production configuration is correct. It should be noted here that although identical input values are used in the SCF and the DAAC, differences in hardware configuration or compilers may produce slightly different output values. This test verifies that operational transfer of the algorithm can be done successfully and that system related tests can be performed. For this test, representative science software will be run on multiple platforms, based on test plans provided with the science software delivery. The upgraded software will be run operationally as an Engineering version and labeled as "Unverified". Product output from the engineering version will be compared to the product output from the existing production version. Status reports are updated, performance statistics are logged and summary reports of the acceptance testing are logged to be used for evaluating the promotion of the science software to the operational phase.

Test Configuration

Hardware–Representative DAAC host (symmetrical multiprocessor or workstation cluster, magnetic storage device)

Software–AITTL CI, Processing CI, SDP Toolkit CI File comparison tools, Science software (algorithm)

Data–Simulated Level 0 and ancillary data

Test Support Required

Data comparison tools, Science software (algorithm), Report generation tools (word processor, spreadsheet, drawing tools)

Test Input

Inputs to this test are science software algorithms that are POSIX, ANSI and ESDIS compliant and have passed the success criteria for evaluation, inspection and verification.

Expected Test Outputs

Outputs to this test include a verified algorithm promoted for production processing, status messages, algorithm output products, acceptance testing summary report, and status logs.

Success Criteria

For each software algorithm submitted, the same shall be executed in the DAAC configuration and promoted to the operational phase.

4.2 TRMM Interface Tests

The following subsections include the threads and builds identified to support testing of TRMM/ ECS interfaces. TRMM Interface testing includes the following threads and builds:

- TRMM Check Thread
- TRMM Ingest Thread
- TRMM Interface Build

4.2.1 TRMM Check Thread (TS004)

This thread demonstrates the ability to setup a network connection between certain TRMM data centers (Sensor Data Processing Facility (SDPF) and TRMM Science Data and Information System (TSDIS)) and ECS and the ability to exchange messages via the network connection in support of data ingest. A connection is initiated by the data provider and establishes communications between ECS and the data provider. Once a connection is established, certain authentication checks are performed to determine if the data provider is authorize to ingest data. If the authentication checks verify the source as invalid the connection is rejected. If the authentication checks verify the source as valid the connection is accepted. Once authenticated the data provider is allowed to send more data messages to ECS and receive responses to data messages.

4.2.1.1 Test Case 1: SDPF Authentication Request with Valid ID Test (TS004.001)

Test Case Description

This test demonstrates the ability of the ECS at the LaRC and MSFC DAACs to receive a valid Authentication Request from the Sensor Data Processing Facility (SDPF) and correctly verify if the request is sent from an approved authorized source. Upon successfully establishing a communications connection, SDPF sends an Authentication Request to ECS. The request is a formatted message sent using TCP. The request includes identification (ID) data. ECS receives the Authentication Request and determines that the originator of the request is authorized by checking the ID data. A response is sent using TCP to SDPF indicating the Authentication Request as valid and therefore the established connection is accepted.

Test Configuration

Hardware– ingest workstation, client host

Software –Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications, CSMS Error Status Logging

Data–N/A.

Test Support Required

Simulator for SDPF interface and the X-Runner Tool to record the test.

Test Input

A series of Authentication Requests in correct SFDU (Standard Formatted Data Unit) format with valid IDs.

Expected Test Outputs

Responses indicating acceptance of the Authentication Requests are logged.

Success Criteria

A communications connection is successfully established and all Authentication Requests sent by SDPF are received and correctly determined to have valid IDs. Each Authentication Request is answered by a response indicating the acceptance of the request and therefore acceptance of the connection.

4.2.1.2 Test Case 2: TSDIS Authentication Requests with Valid ID Test (TS004.002)

Test Case Description

This test demonstrates the ability of the ECS at GSFC and MSFC DAACs to receive a valid Authentication Request from TRMM Science Data and Information System (TSDIS) and correctly verify if the request is sent from an approved authorized source. Upon successfully establishing a communications connection, SDPF sends an Authentication Request to ECS. The request is a formatted message sent using TCP. The request includes identification (ID) data. ECS receives the Authentication Request and determines that the originator of the request is authorized by checking the ID data. A response is sent using TCP to TSDIS indicating the Authentication Request is valid and therefore the established connection is accepted.

Test Configuration

Hardware–ingest workstation, client host

Software–Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications, CSMS Error Status Logging

Data–N/A.

Test Support Required

Simulator for TSDIS interface and the X-Runner Tool to record the test.

Test Input

A series of Authentication Requests in correct SFDU (Standard Formatted Data Unit) format with valid IDs.

Expected Test Outputs

Responses indicating acceptance of the Authentication Requests are logged.

Success Criteria

A communications connection is successfully established and all Authentication Requests sent by TSDIS are received and correctly determined to have valid IDs. Each Authentication Request is answered by a response indicating the acceptance of the request and therefore acceptance of the connection.

4.2.1.3 Test Case 3: SDPF Authentication Requests with Invalid ID Test (TS004.003)

Test Case Description

This test demonstrates the ability of the ECS at LaRC and MSFC DAACs to receive and recognize an invalid Authentication Request from SDPF and correctly verify that the request is sent from an unauthorized source. Upon successfully establishing a communications connection, SDPF sends an Authentication Request to ECS. The request is a formatted message sent using TCP. The request includes invalid identification (ID) data. ECS receives the Authentication Request and determines that the originator of the request is not authorized by checking the ID data. A response is sent using TCP to SDPF indicating the Authentication Request is invalid and therefore the established connection is rejected.

Test Configuration

Hardware– ingest workstation, client host

Software–Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications, CSMS Error Status Logging

Data–N/A.

Test Support Required

Simulator for SDPF interface and the X-Runner Tool to record the test.

Test Input

A series of Authentication Requests in correct SFDU (Standard Formatted Data Unit) format with IDs not recognized by the DAAC as that of a valid user.

Expected Test Outputs

Responses indicating rejection of the Authentication Requests are logged.

Success Criteria

A communications connection is successfully established and all Authentication Requests sent by SDPF are received and correctly determined to have invalid IDs. Each Authentication Request is answered by a response indicating the rejection of the request and therefore rejection of the connection.

4.2.1.4 Test Case 4: TSDIS Authentication Request with Invalid ID Test (TS004.004)

Test Case Description

This test demonstrates the ability of the ECS at GSFC and MSFC DAACs to receive and recognize an invalid Authentication Request from TSDIS and correctly verify that the request is sent from an unauthorized source. Upon successfully establishing a communications connection, TSDIS sends an Authentication Request to ECS. The request is a formatted message sent using TCP. The request includes invalid identification (ID) data. ECS receives the Authentication Request and determines that the originator of the request is not authorized by checking the ID data. A response is sent using TCP to TSDIS indicating the Authentication Request is invalid and therefore the established connection is rejected.

Test Configuration

Hardware–ingest workstation, client host

Software– Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications, CSMS Error Status Logging

Data–N/A.

Test Support Required

Simulator for TSDIS interface and the X-Runner Tool to record the test.

Test Input

A series of Authentication Requests in correct SFDU (Standard Formatted Data Unit) format with IDs not recognized by the DAAC as that of a valid user.

Expected Test Outputs

Responses indicating rejection of the Authentication Requests are logged.

Success Criteria

A communications connection is successfully established and all Authentication Requests sent by TSDIS are received and correctly determined to have invalid IDs. Each Authentication Request is answered by a response indicating the rejection of the request and therefore rejection of the connection.

4.2.1.5 Test Case 5: SDPF Valid Data Availability Notice Verification Test (TS004.005)

Test Case Description

This test demonstrates the ability to receive Data Availability Notices from SDPF and validate the notices for adherence to ECS standards. Data ingest is initiated when the data provider sends a Data Availability Notice (DAN) informing ECS that data is available for ingest. A DAN contains a header, linked to data files. The header contains information about the linked files.

The linked data files describe a data product using CCSDS standards. One DAN may describe one data file, or several data files, that are available for ingest. ECS does certain validation checks on the DAN, to determine that required fields are present and that the format of the message (ie data/time) is correct and consistent with the standards. After validation of the DAN a DAN Acknowledgment (DAA) message is sent to SDPF.

Test Configuration

Hardware– ingest workstation, client host

Software–Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications, CSMS Error Status Logging

Data–N/A

Test Support Required

Simulator for SDPF interface and the X-Runner Tool to record the test.

Test Input

Inputs to this test include a series of DANs submitted electronically using a SDPF simulated interface to ECS. This includes DANs with a single detached header linked to a single file and DANs with single detached headers linked to multiple files. Only valid DANs are submitted.

Expected Test Outputs

Outputs to this test include a DAA for each DAN received.

Success Criteria

This test is deemed successful if each DAN submitted is received and validated correctly, and a DAA is sent to SDPF indicating the DAN as valid.

4.2.1.6 Test Case 6: TSDIS Valid Data Availability Notice Verification Test (TS004.006)

Test Case Description

This test demonstrates the ability for the ECS at GSFC and the MSFC DAACs to receive Data Availability Notices from TSDIS and validate the notices for adherence to ECS standards. Data ingest is initiated when the data provider sends a Data Availability Notice (DAN) informing ECS that data is available for ingest. A DAN contains a header, linked to data files. The header contains information about the linked files. The linked data files describe a data product using CCSDS standards. One DAN may describe one data file, or several data files, that are available for ingest. ECS does certain validation checks on the DAN, to determine that required fields are present and that the format of the message (ie data/time) is correct and consistent with the standards. After validation of the DAN a Data Availability Acknowledgment (DAA) is sent to TSDIS.

Test Configuration

Hardware–ingest workstation, client host

Software–Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications, CSMS Error Status Logging

Data–N/A

Test Support Required

Simulator for TSDIS interface and X-Runner Tool to record the test.

Test Input

Inputs to this test include a series of DANs submitted electronically using a TSDIS simulated interface to ECS. This includes DANs with a single detached header linked to a single file and DANs with single detached headers linked to multiple files. Only valid DANs are submitted.

Expected Test Outputs

Outputs to this test include a DAA for each DAN received.

Success Criteria

This test is deemed successful if each DAN submitted is received and validated correctly, and a DAA is sent to TSDIS, indicating the DAN as valid.

4.2.1.7 Test Case 7: SDPF Invalid Data Availability Notice Verification Test (TS004.007)

Test Case Description

This test demonstrates the ability to recognize Data Availability Notices from SDPF, that do not adhere to ECS standards, as invalid. Data ingest is initiated when the data provider sends a Data Availability Notice (DAN) informing ECS that data is available for ingest. A DAN contains a header, linked to data files. The header contains information about the linked files. The linked data files describe a data product using CCSDS standards. One DAN may describe one data file, or several data files, that are available for ingest. ECS validates the DAN, checking that all required fields are present and that the format of the message is correct and consistent with the standards. For this test all DANs submitted are invalid. The ECS determines the DAN is invalid and a message is sent to SDPF, indicating the disposition of the DAN as invalid. The message indicates the reason for declaring a DAN as invalid. If a DAN is deemed invalid on account of invalid file parameters, only those files with invalid parameters are reported.

Test Configuration

Hardware–ingest workstation, client host

Software–Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications, CSMS Error Status Logging

Data– N/A

Test Support Required

Simulator for SDPF interface and the X-Runner Tool to record the test.

Test Input

Inputs to this test include a series of erroneous DANs submitted electronically using a SDPF simulated interface to ECS. This includes DANs with a single detached header linked to a single file and DANs with single detached headers linked to multiple files. Only invalid DANs are submitted. At least one invalid DAN is submitted for each possible error condition validated by ECS according to ECS defined standards.

Expected Test Outputs

Outputs to this test include status messages sent from ECS to the data provider for each DAN received.

Success Criteria

This test is deemed successful if each DAN submitted is received and validated correctly, and a status message is sent in response, indicating the DAN as invalid. For DANs with multiple files, each invalid file is indicated in the message.

4.2.1.8 Test Case 8: TSDIS Invalid Data Availability Notice Verification Test (TS004.008)

Test Case Description

This test demonstrates the ability to recognize Data Availability Notices from TSDIS that do not adhere to ECS standards as invalid. Data ingest is initiated when the data provider sends a Data Availability Notice (DAN) informing ECS that data is available for ingest. A DAN contains a header, linked to data files. The header contains information about the linked files. The linked data files describe a data product using CCSDS standards. One DAN may describe one data file, or several data files, that are available for ingest. ECS validates the DAN, checking that all required fields are present and that the format of the message is correct and consistent with the standards. For this test all DANs submitted are invalid. The ECS determines the DAN is invalid and a status message is sent to SDPF, indicating the disposition of the DAN as invalid. The message indicates the reason for declaring a DAN as invalid. For DANs with multiple files, each invalid file is indicated.

Test Configuration

Hardware–ingest workstation, client host

Software–Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications, CSMS Error Status Logging

Data–N/A

Test Support Required

Simulator for TSDIS interface the X-Runner Tool to record the test.

Test Input

Inputs to this test include a series of erroneous DANs submitted electronically using a TSDIS simulated interface to ECS. This includes DANs with a single detached header linked to a single file and DANs with single detached headers linked to multiple files. Only invalid DANs are submitted. At least one invalid DAN is submitted for each possible error condition validated by ECS according to ECS defined standards.

Expected Test Outputs

Outputs to this test include status messages sent from ECS to the data provider for each DAN received.

Success Criteria

This test is deemed successful if each DAN submitted is received and validated correctly, and status is sent to SDPF, indicating the DAN as invalid. For DANs with multiple files, each invalid file is indicated.

4.2.2 TRMM Ingest Thread (TS005)

This thread demonstrates acceptance of TRMM data for storage by ECS. For SDPF and TSDIS data interfaces, ingest is initiated by a Data Availability Notice (DAN) sent by the data provider. The data is transferred by the DAAC and placed in ECS storage. For NESDIS and GDAO interfaces, ingest is initiated once data availability is determined using a polling mechanism.

4.2.2.1 Test Case 1: SDPF FTP-Get Single File Data Ingest Test (TS005.001)

Test Case Description

This test demonstrates the ability to ingest a data collection containing a single data file of SDPF data. A Data Availability Notice (DAN) is sent from a simulated SDPF interface and is received by ECS, indicating the availability of data for ingest. An interim capability for file transfer is available for early interface file transfer testing. The file indicated in the DAN is retrieved via FTP. The data is placed on temporary magnetic storage.

Test Configuration

Hardware–ingest workstation, client host, storage device and media

Software–Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for IR1 only), CSMS Interprocess Communications, CSMS Error Status Logging

Data–simulated CERES and LIS L0 data.

Test Support Required

Simulator for SDPF interface and the X-Runner Tool to record the test.

Test Input

Inputs to this test include: Data Availability Notices, at least one data file for each data type to include CERES and LIS L0 data sets.

Expected Test Outputs

Outputs to this test include successful transfer of data.

Success Criteria

This test is deemed successful if each DAN submitted from SDPF is successfully received at ECS. All DANs received, result in successful retrieval of data from SDPF to ECS. Data comparison of the data before ingest to data after ingest, shows no significant differences.

4.2.2.2 Test Case 2: TSDIS FTP-Get Single File Data Ingest Test (TS005.002)

Test Case Description

This test demonstrates the ability to ingest a data collection containing a single data file of TSDIS data. A Data Availability Notice (DAN) is sent from a simulated TSDIS interface and is received by ECS, indicating the availability of data for ingest. An interim capability for file transfer is available for early interface file transfer testing. The file indicated in the DAN is retrieved via FTP. The data is placed on temporary magnetic storage.

Test Configuration

Hardware– ingest workstation, client host, storage device and media

Software–Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for IR1 only), CSMS Interprocess Communications, CSMS Error Status Logging

Data–simulated TSDIS data.

Test Support Required

Simulator for TSDIS interface and the X-Runner Tool to record the test.

Test Input

Inputs to this test include: Data Availability Notices, and at least one data file for each data type to include all ECS stored TSDIS data types.

Expected Test Outputs

Outputs to this test include successful transfer of data.

Success Criteria

This test is deemed successful if each DAN submitted from TSDIS is successfully received at ECS. All DANs received, result in successful retrieval of data from TSDIS to ECS. Data comparison of the data before ingest to data after ingest, shows no significant differences.

4.2.2.3 Test Case 3: SDPF FTP-Get and Multiple File Ingest Test (TS005.003)

Test Case Description

This test case demonstrates the ability to ingest multiple SDPF data files. A Data Availability Notice is received by ECS. The DAN contains a header, linked to a product specification for multiple data files. The DAN contains data set identification, and data granule identification. An interim capability for file transfer is available for early interface file transfer testing. Data is transferred and placed in an ECS directory.

Test Configuration

Hardware– ingest workstation, client host, storage device and media

Software–Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for IR1 only), CSMS Interprocess Communications, CSMS Error Status Logging

Data–simulated CERES and LIS L0 data.

Test Support Required

Simulator for SDPF interface and the X-Runner Tool to record the test.

Test Input

Inputs to this test include: Data Availability Notices, and at least one data file for each SDPF data type for multiple file ingest.

Expected Test Outputs

Outputs to this test include successful delivery of DANs and data for ingest.

Success Criteria

This test is deemed successful if each DAN submitted from SDPF is successfully received at ECS. All DANs received, result in successful delivery of data from SDPF to ECS. Data comparison of the data before ingest to data after ingest, shows no significant differences.

4.2.2.4 Test Case 4: TSDIS FTP-Get and Multiple File Ingest Test (TS005.004)

Test Case Description

This test case demonstrates the ability to ingest multiple TSDIS data files. A Data Availability Notice is received by ECS. The DAN contains a header, linked to a product specification for multiple data files. The DAN contains data set identification, and data granule identification. An interim capability for file transfer is available for early interface file transfer testing. Data is transferred and placed in an ECS directory.

Test Configuration

Hardware– ingest workstation, client host, storage device and media

Software–Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for IR1 only), CSMS Interprocess Communications, CSMS Error Status Logging

Data–simulated TSDIS data.

Test Support Required

Simulator for TSDIS interface and the X-Runner Tool to record the test.

Test Input

Inputs to this test include: Data Availability Notices, and at least one data file for each TSDIS data type for multiple file ingest.

Expected Test Outputs

Outputs to this test include successful delivery of DANs and data for ingest.

Success Criteria

This test is deemed successful if each DAN submitted from TSDIS is successfully received at ECS. All DANs received, result in successful delivery of data from TSDIS to ECS. Data comparison of the data before ingest to data after ingest, shows no significant differences.

4.2.2.5 Test Case 5: NESDIS and GDAO Polling Test (TS005.005)

Test Case Description

This test case demonstrates the ability to initiate ingest for TRMM ancillary data files. A polling client periodically checks a remote location for the presence of data granule files. Upon detection of a data granule a Polling Ingest Request is generated.

Test Configuration

Hardware–ingest workstation, client host, storage device and media

Software–Ingest Client Interface CSC, Polling Ingest Client Interface CSC, CSMS Interprocess Communications, CSMS Error Status Logging

Data–simulated NESDIS and GDAO ancillary data.

Test Support Required

Ancillary data in a location accessible to the ESN and the X-Runner Tool to record the test.

Test Input

Inputs to this test include detection of TRMM ancillary data files.

Expected Test Outputs

Outputs to this test include entries into a log indicating generation of a Polling Request.

Success Criteria

This test is deemed successful if each file detected as a result of polling, is entered in the log.

4.2.3 TRMM Interface Build (BS002)

The TRMM Interface Build integrates the functionality of the TRMM Data Check Thread, and the TRMM Data Ingest Thread. Two mechanisms are used for data ingest. One mechanism supports process communications including data message passing for data ingest. The second mechanism involves using a polling mechanism to support data transfer for data ingest. This build verifies the ability to initiate data communications for those interfaces which support process communications (TSDIS and SDPF) which results in data transfer and ingest. Also tested is data ingest of TRMM ancillary data (NESDIS and GDAO interfaces) using the polling mechanism for data transfer and data ingest. Included are tests which verify responses to error conditions including: file transfer failure, file size discrepancies, invalid data type identifier, missing required metadata or metadata with out of range parameters, failed data conversion, failed archiving of data, and the inability to transfer data in a specified amount of time.

4.2.3.1 Test Case 1: SDPF FTP-Get File Validation and Ingest Test (BS002.001)

Test Case Description

This test demonstrates the ability to establish a network connection between SDPF and ECS for ingest of a series of data collections, including collections containing single and multiple granules of data. SDPF sends an Authentication Request to ECS. ECS verifies the Authentication Request as from a valid source and a connection is established allowing SDPF to send Data Availability Notices. The DANs are received and validated. Data Availability Acknowledgments (DAAs) are sent to SDPF. Upon validation, the data is retrieved from SDPF. Data is placed in ECS storage.

Test Configuration

Hardware–ingest workstation, client host, storage device and media

Software–Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software only for IR1), CSMS Interprocess Communications, CSMS Error Status Logging

Data–Simulated CERES and LIS L0 data.

Test Support Required

Simulator for SDPF interface, data comparison tools, and the X-Runner Tool to record the test.

Test Input

Inputs to this test include: Data Availability Notices, at least one data file for each data type to include CERES and LIS L0 and processed data for single and multiple file ingest.

Expected Test Outputs

Outputs to this test include Data Availability Acknowledgments (DAAs) which notify the data provider that the DAN is received and validated and Data Delivery Notices (DDNs) indicating successful transfer of data.

Success Criteria

A data connection is successfully established allowing SDPF to send Data Availability Notices to ECS. Data acknowledgment for each DAN is sent to SDPF. For each DAN, the data is successfully retrieved and placed on storage media. Data Delivery Notices are sent to SDPF. Data comparison of the data before ingest to data after ingest, shows no significant differences.

4.2.3.2 Test Case 2: TSDIS FTP-Get File Validation and Ingest Test (BS002.002)

Test Case Description

This test demonstrates the ability to establish a network connection between TSDIS and ECS for ingest of a series of data collections, including collections containing single and multiple granules of data. TSDIS sends an Authentication Request to ECS. ECS verifies the Authentication Requests as from a valid source and a connection is established allowing TSDIS to send Data Availability Notices. The DANs are received and validated. Data Availability Acknowledgments (DAAs) are sent to TSDIS. Upon validation, the data is retrieved from TSDIS and placed on storage media.

Test Configuration

Hardware–ingest workstation, client host, storage device and media

Software–Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for IR1 only), CSMS Interprocess Communications, CSMS Error Status Logging

Data–Simulated PR, TMI, GV and VIRS data.

Test Input

Inputs to this test include: Data Availability Notices, at least one data file for each data type to include TSDIS data for single and multiple file ingest, and Data Delivery Acknowledgments (DDAs).

Expected Test Outputs

Outputs to this test include Data Availability Acknowledgments (DAAs) which notify the data provider that the DAN is received and validated and Data Delivery Notices indicating successful transfer of data.

Success Criteria

A data connection is successfully established allowing TSDIS to send Data Availability Notices to ECS. Data acknowledgment for each DAN is sent to TSDIS. For each DAN, the data is successfully retrieved and ingested. Data is placed on storage media. Data Delivery notices are sent to TSDIS. Data comparison of the data before ingest to data after ingest, shows no significant differences.

4.2.3.3 Test Case 3: SDPF Status Reporting Test (BS002.003)

Test Case Description

This test demonstrates the ability to send status messages to SDPF. A status message is generated by ECS and sent to SDPF. Status messages are required for the following:

- file transfer failure
- file size discrepancies
- invalid data type identifier
- missing required metadata
- metadata parameters out of range
- data conversion failure
- failure to archive data
- inability to transfer data in a specified time window
- missing required request information
- successful archive of data

Test Configuration

Hardware– workstation, terminal, ingest client host

Software–Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for IR1 only), CSMS Interprocess Communications, CSMS Error Status Logging

Data–simulated TRMM data.

Test Support Required

Simulator for SDPF interface, and the X-Runner Tool to record the test.

Test Input

Inputs to this test include ingest requests. At least one ingest request is submitted to test for each instance of status reporting.

Expected Test Outputs

Outputs to this test include correct status messages to the ingest request provider.

Success Criteria

This test is deemed successful if all ingest requests submitted are received and for each ingest request received an appropriate status message is sent to SDPF.

4.2.3.4 Test Case 4: TSDIS Status Reporting Test (BS002.004)

Test Case Description

This test demonstrates the ability to send status messages to TSDIS. A status message is generated by ECS and sent to TSDIS. Status messages are required for the following:

- file transfer failure
- file size discrepancies
- invalid data type identifier
- missing required metadata
- metadata parameters out of range
- data conversion failure
- failure to archive data
- inability to transfer data in a specified time window
- missing required request information
- successful archive of data

Test Configuration

Hardware–workstation, terminal, ingest client host

Software –Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for IR1 only), CSMS Interprocess Communications, CSMS Error Status Logging

Data–simulated TRMM data.

Test Support Required

Simulator for TSDIS interface, and the X-Runner Tool to record the test.

Test Input

Inputs to this test include ingest requests. At least one ingest request is submitted to test for each instance of status reporting.

Expected Test Outputs

Outputs to this test include correct status messages to the ingest request provider.

Success Criteria

This test is deemed successful if all ingest requests submitted are received and for each ingest request received an appropriate status message is sent to TSDIS.

4.2.3.5 Test Case 5: NESDIS and GDAO Polling Ingest Test (BS002.005)

Test Case Description

This test case demonstrates the ability to initiate ingest by polling a location for the presence of TRMM ancillary data files and upon detection of a data granule generate a Polling Request. This Polling Request indicates the location of the data files. The data files identified in the Polling Request are transferred and the data is placed in ECS storage. Data comparison of the data before ingest to data after ingest, shows no significant differences.

Test Configuration

Hardware–workstation, terminal, ingest client host, storage devices and media

Ingest Client Interface CSC, Polling Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for IR1 only), CSMS Interprocess Communications, CSMS Error Status Logging

Data–simulated NESDIS and GDAO ancillary data.

Test Support Required

Ancillary data in a location accessible to the ESN and the X-Runner Tool.

Test Input

Inputs to this test include detection and transfer of TRMM ancillary data files.

Expected Test Outputs

Outputs to this test include entries into a log indicating generation of a Polling Request.

Success Criteria

This test is deemed successful if files are detected as a result of polling, and an Ingest Polling Request is generated. Each Polling Request is assigned a unique id and contains the location of the data. Data is successfully transferred. Data comparison of the data before ingest to data after ingest, shows no significant differences.

This page intentionally left blank.

Appendix A: Test Tool Requirements

The following is a list of test tools needed to support IR1 SDPS I&T.

1. File Comparison Tools- A file comparison utility is needed to compare data outputs with data inputs or to compare data outputs run under different configurations. This is in support of the following test situations:

   - Science Software testing comparing test outputs produced when running test scripts with the SCF version of the SDP Toolkit (in the SCF environment) against expected test outputs included in a science software delivery
   - Science Software testing comparing test outputs produced when running test scripts with the SCF version of the SDP Toolkit (in the DAAC environment) against expected test outputs included in a science software delivery
   - Science Software testing comparing test outputs produced when running test scripts with the DAAC version of the SDP Toolkit (in the DAAC environment) against expected test outputs included in a science software delivery
   - TRMM ingest interface testing comparing files before ingest and after transfer from the TSDIS and SDPF interfaces.

2. Reporting Tools - A mechanism is needed to capture test results for analysis. This is in support of the following test situation:

   - Recording of integration status throughout all phases of Science Software integration (e.g. plotting tools, spreadsheets, editors, drawing tools)
   - Logging of ingest status messages and error messages for off-line review.

3. Interface Simulators - Interface simulators are tools that pass messages between ECS and an external interface. The interface is not available at the time of testing. The following tools are needed for IR1 I&T testing:

   - SDPF simulator for message passing between ECS and SDPF for data ingest testing. The messages include: Data Authentication Requests, Data Authentication Responses, Data Availability Notices, Data Availability Acknowledgments, Data Delivery Notices, and Data Delivery Acknowledgments.
   - TSDIS simulator for message passing between ECS and TSDIS for data ingest testing. The messages include: Data Authentication Requests, Data Authentication Responses, Data Availability Notices, Data Availability Acknowledgments, Data Delivery Notices, and Data Delivery Acknowledgments.

This page intentionally left blank.

Appendix B: Verification Traceability Matrix

The following is a matrix mapping level 4 requirements to test cases.

Test Case #	Test Case Name	L4 Requirement #
TS001.001	SCF Toolkit in Native Environment Test	S-DPS-70010
TS001.001	SCF Toolkit in Native Environment Test	S-DPS-70020
TS001.001	SCF Toolkit in Native Environment Test	S-DPS-70030
TS001.002	SCF Toolkit in DAAC Environment Test	S-DPS-70010
TS001.002	SCF Toolkit in DAAC Environment Test	S-DPS-70020
TS001.002	SCF Toolkit in DAAC Environment Test	S-DPS-70030
TS001.003	DAAC Toolkit in DAAC Environment Test	S-DPS-70010
TS001.003	DAAC Toolkit in DAAC Environment Test	S-DPS-70020
TS001.003	DAAC Toolkit in DAAC Environment Test	S-DPS-70030
TS002.001	Science Software (Algorithm) Delivery Verification Test	S-DPS-40010
TS002.002	Science Software (Algorithm) Delivery Verification Test	S-DPS-40100
TS002.002	Science Software (Algorithm) Delivery Verification Test	S-DPS-40110
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40210
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40220
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40220-a
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40220-b
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40220-c
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40220-d
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40220-e
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40220-f
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40220-g
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40220-h
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40220-i
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40220-j
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40220-k
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40220-l
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40220-m
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40320
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40310
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40310-a
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40310-b
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40310-c
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40310-d
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40310-e

Test Case #	Test Case Name	L4 Requirement #
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40310-f
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40310-g
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40310-h
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40310-i
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40310-j
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40310-k
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40260
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40270
TS002.003	Code Standard Checking Test - FORTRAN 77 Code	S-DPS-40340
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40320
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40310
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40310-a
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40310-b
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40310-c
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40310-d
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40310-e
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40310-f
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40310-g
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40310-h
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40310-i
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40310-j
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40310-k
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40260
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40270
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40340
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40230
TS002.004	Code Standard Checking Test - FORTRAN 90 Code	S-DPS-40240
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40320
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40200
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40290
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40290-a
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40300
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40300-a
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40300-b
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40300-c
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40300-d
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40300-e
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40300-f
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40300-g
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40300-h

Test Case #	Test Case Name	L4 Requirement #
TS002.005	Code Standard Checking Test - “C” Code	S-DPS-40300-i
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40300-j
TS002.005	Code Standard Checking Test - "C" Code	S-DPS-40300-k
TS002.006	Code Standard Checking Test - Ada Code	S-DPS-40320
TS002.006	Code Standard Checking Test - Ada Code	S-DPS-40250
TS002.007	Code Standard Checking Test - SDP Toolkit Usage Requirements	S-DPS-40280
TS002.007	Code Standard Checking Test - SDP Toolkit Usage Requirements	S-DPS-40280-a
TS002.007	Code Standard Checking Test - SDP Toolkit Usage Requirements	S-DPS-40280-b
TS002.007	Code Standard Checking Test - SDP Toolkit Usage Requirements	S-DPS-40280-c
TS002.007	Code Standard Checking Test - SDP Toolkit Usage Requirements	S-DPS-40280-d
TS002.007	Code Standard Checking Test - SDP Toolkit Usage Requirements	S-DPS-40280-e
TS002.007	Code Standard Checking Test - SDP Toolkit Usage Requirements	S-DPS-40280-f
TS002.007	Code Standard Checking Test - SDP Toolkit Usage Requirements	S-DPS-40280-g
TS002.008	Code Standard Checking - Extract Required Header Data Test	S-DPS-40330
TS002.009	Development Environment Test	S-DPS-40500
TS002.009	Development Environment Test	S-DPS-40510
TS002.009	Development Environment Test	S-DPS-40520
TS002.009	Development Environment Test	S-DPS-40530
TS002.009	Development Environment Test	S-DPS-40540
TS002.009	Development Environment Test	S-DPS-40550
TS002.009	Development Environment Test	S-DPS-40560
TS002.009	Development Environment Test	S-DPS-40570
TS002.009	Development Environment Test	S-DPS-40580
TS002.009	Development Environment Test	S-DPS-41630
TS002.009	Development Environment Test	S-DPS-41640
TS002.009	Development Environment Test	S-DPS-41700
TS002.009	Development Environment Test	S-DPS-41620
TS002.010	File Comparison Test	S-DPS-40900
TS002.010	File Comparison Test	S-DPS-40910
TS002.010	File Comparison Test	S-DPS-40920
TS002.010	File Comparison Test	S-DPS-40930
TS002.010	File Comparison Test	S-DPS-40940
TS002.011	Configuration Management Test	S-DPS-41400
TS002.011	Configuration Management Test	S-DPS-41410
TS002.012	Status Tracking and Report Generation Test	S-DPS-41500
TS002.012	Status Tracking and Report Generation Test	S-DPS-41510
TS002.012	Status Tracking and Report Generation Test	S-DPS-41520
TS002.012	Status Tracking and Report Generation Test	S-DPS-41530
TS002.012	Status Tracking and Report Generation Test	S-DPS-42740

Test Case #	Test Case Name	L4 Requirement #
TS002.012	Status Tracking and Report Generation Test	S-DPS-42750
TS002.012	Status Tracking and Report Generation Test	S-DPS-42760
TS002.013	Operating System and Utilities Test	S-DPS-41700
TS002.013	Operating System and Utilities Test	S-DPS-41620
TS002.013	Operating System and Utilities Test	S-DPS-41600
TS002.013	Operating System and Utilities Test	S-DPS-41610
TS002.013	Operating System and Utilities Test	S-DPS-41650
TS002.014	Internet Utilities Test	S-DPS-41800
TS002.014	Internet Utilities Test	S-DPS-41810
TS002.014	Internet Utilities Test	S-DPS-41820
TS002.014	Internet Utilities Test	S-DPS-41830
TS002.014	Internet Utilities Test	S-DPS-41840
TS002.014	Internet Utilities Test	S-DPS-42640
TS002.014	Internet Utilities Test	S-DPS-42710
TS002.015	Product Metadata Test	S-DPS-42000
TS002.015	Product Metadata Test	S-DPS-42010
TS002.016	Profiling Test	S-DPS-41000
TS002.16	Profiling Test	S-DPS-41005
TS002.016	Profiling Test	S-DPS-41010
TS002.016	Profiling Test	S-DPS-41015
TS002.016	Profiling Test	S-DPS-41020
TS002.016	Profiling Test	S-DPS-41030
TS002.016	Profiling Test	S-DPS-41035
TS002.016	Profiling Test	S-DPS-41040
TS003.001	Manual Submittal of Data Processing Requests	TBD
TS003.002	Process Queue	TBD
TS003.003	PGE Execution	TBD
TS003.004	Status of Data Processing Request	TBD
TS003.004	Status of Data Processing Request	S-DPS-60970
TS003.005	Processing System Initialization and Shutdown	S-DPS-60080
TS003.005	Processing System Initialization and Shutdown	S-DPS-60120
TS003.006	Science Processing Operating System and Utilities	S-DPS-61110
TS003.006	Science Processing Operating System and Utilities	S-DPS-61120
TS003.006	Science Processing Operating System and Utilities	S-DPS-61130
TS003.006	Science Processing Operating System and Utilities	S-DPS-61140
TS003.006	Science Processing Operating System and Utilities	S-DPS-61150
TS003.006	Science Processing Operating System and Utilities	S-DPS-61160
TS003.007	Science Processing Development Environment	S-DPS-61171
TS003.007	Science Processing Development Environment	S-DPS-61172

Test Case #	Test Case Name	L4 Requirement #
TS003.007	Science Processing Development Environment	S-DPS-61173
TS003.007	Science Processing Development Environment	S-DPS-61174
TS003.007	Science Processing Development Environment	S-DPS-61175
TS003.007	Science Processing Development Environment	S-DPS-61177
TS003.008	Science Processing Documentation	S-DPS-61170
TS004.001	SDPF Authentication Request with Valid ID Test	S-INS-00030
TS004.001	SDPF Authentication Request with Valid ID Test	S-INS-00040
TS004.002	TSDIS Authentication Requests with Valid ID Test	S-INS-00030
TS004.002	TSDIS Authentication Requests with Valid ID Test	S-INS-00040
TS004.003	SDPF Authentication Requests with Invalid ID Test	S-INS-00030
TS004.003	SDPF Authentication Requests with Invalid ID Test	S-INS-00040
TS004.004	TSDIS Authentication Request with Invalid ID Test	S-INS-00030
TS004.004	TSDIS Authentication Request with Invalid ID Test	S-INS-00040
TS004.005	SDPF Valid Data Availability Notice Verification Test	S-INS-00010
TS004.005	SDPF Valid Data Availability Notice Verification Test	S-INS-00020
TS004.005	SDPF Valid Data Availability Notice Verification Test	S-INS-00060h
TS004.005	SDPF Valid Data Availability Notice Verification Test	S-INS-00060i
TS004.006	TSDIS Valid Data Availability Notice Verification Test	S-INS-00010
TS004.006	TSDIS Valid Data Availability Notice Verification Test	S-INS-00020
TS004.006	TSDIS Valid Data Availability Notice Verification Test	S-INS-00060h
TS004.006	TSDIS Valid Data Availability Notice Verification Test	S-INS-00060i
TS004.007	SDPF Invalid Data Availability Notice Verification Test	S-INS-00060
TS004.008	TSDIS Invalid Data Availability Notice Verification Test	S-INS-00060
TS005.001	SDPF FTP-Get Single File Data Ingest Test	S-INS-00415
TS005.001	SDPF FTP-Get Single File Data Ingest Test	S-INS-00520
TS005.001	SDPF FTP-Get Single File Data Ingest Test	S-INS-00540
TS005.002	TSDIS FTP-Get Single File Data Ingest Test	S-INS-00415
TS005.002	TSDIS FTP-Get Single File Data Ingest Test	S-INS-00560
TS005.002	TSDIS FTP-Get Single File Data Ingest Test	S-INS-00570
TS005.003	SDPF FTP-Get and Multiple File Ingest Test	S-INS-00415
TS005.003	SDPF FTP-Get and Multiple File Ingest Test	S-INS-00520
TS005.003	SDPF FTP-Get and Multiple File Ingest Test	S-INS-00540
TS005.004	TSDIS FTP-Get and Multiple File Ingest Test	S-INS-00415
TS005.004	TSDIS FTP-Get and Multiple File Ingest Test	S-INS-00560
TS005.004	TSDIS FTP-Get and Multiple File Ingest Test	S-INS-00570
TS005.005	NESDIS and GDAO Polling Test	S-INS-00100
TS005.005	NESDIS and GDAO Polling Test	S-INS-00110
BS001.001	Ingestion of Delivery Package	S-DPS-40010
BS001.001	Ingestion of Delivery Package	S-DPS-42100

Test Case #	Test Case Name	L4 Requirement #
BS001.001	Ingestion of Delivery Package	S-DPS-42200
BS001.002	Evaluation, Inspection and Verification of the Science Software	S-DPS-42110
BS001.002	Evaluation, Inspection and Verification of the Science Software	S-DPS-42120
BS001.002	Evaluation, Inspection and Verification of the Science Software	S-DPS-42130
BS001.002	Evaluation, Inspection and Verification of the Science Software	S-DPS-42140
BS001.002	Evaluation, Inspection and Verification of the Science Software	S-DPS-42150
BS001.002	Evaluation, Inspection and Verification of the Science Software	S-DPS-42160
BS001.002	Evaluation, Inspection and Verification of the Science Software	S-DPS-42170
BS001.002	Evaluation, Inspection and Verification of the Science Software	S-DPS-42180
BS001.002	Evaluation, Inspection and Verification of the Science Software	S-DPS-42190
BS001.002	Evaluation, Inspection and Verification of the Science Software	S-DPS-42700
BS001.002	Evaluation, Inspection and Verification of the Science Software	S-DPS-42770
BS001.003	Science Software Delivery Integration Test	S-DPS-42300
BS001.003	Science Software Delivery Integration Test	S-DPS-42310
BS001.003	Science Software Delivery Integration Test	S-DPS-42320
BS001.003	Science Software Delivery Integration Test	S-DPS-42330
BS001.003	Science Software Delivery Integration Test	S-DPS-42350
BS001.003	Science Software Delivery Integration Test	S-DPS-42360
BS001.003	Science Software Delivery Integration Test	S-DPS-42500
BS001.003	Science Software Delivery Integration Test	S-DPS-42780
BS001.003	Science Software Delivery Integration Test	S-DPS-42700
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42370
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42500
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42510
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42520
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42530
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42540
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42550
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42560
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42570
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42580
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42590
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42600
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42610
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42620
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42630
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42700
BS001.004	Acceptance Testing of the Science Software Delivery	S-DPS-42790
BS002.001	SDPF FTP-Get File Validation and Ingest Test	S-INS-00010

Test Case #	Test Case Name	L4 Requirement #
BS002.002	TSDIS FTP-Get File Validation and Ingest Test	S-INS-00020
BS002.001	SDPF FTP-Get File Validation and Ingest Test	S-INS-00030
BS002.001	SDPF FTP-Get File Validation and Ingest Test	S-INS-00040
BS002.001	SDPF FTP-Get File Validation and Ingest Test	S-INS-00415
BS002.001	SDPF FTP-Get File Validation and Ingest Test	S-INS-00520
BS002.001	SDPF FTP-Get File Validation and Ingest Test	S-INS-00540
BS002.002	TSDIS FTP-Get File Validation and Ingest Test	S-INS-00010
BS002.002	TSDIS FTP-Get File Validation and Ingest Test	S-INS-00020
BS002.002	TSDIS FTP-Get File Validation and Ingest Test	S-INS-00030
BS002.002	TSDIS FTP-Get File Validation and Ingest Test	S-INS-00040
BS002.002	TSDIS FTP-Get File Validation and Ingest Test	S-INS-00415
BS002.002	TSDIS FTP-Get File Validation and Ingest Test	S-INS-00560
BS002.002	TSDIS FTP-Get File Validation and Ingest Test	S-INS-00570
BS002.003	SDPF Status Reporting Test	S-INS-00060
BS002.004	TSDIS Status Reporting Test	S-INS-00060
BS002.005	NESDIS and GDAO Polling Ingest Test	S-INS-00100
BS002.005	NESDIS and GDAO Polling Ingest Test	S-INS-00110
BS002.005	NESDIS and GDAO Polling Ingest Test	S-INS-00415

This page intentionally left blank.

Abbreviations and Acronyms

AI&T	Algorithm Integration and Test
AIT	algorithm integration team
AIT	algorithm integration and test
AM-1	EOS AM Project spacecraft 1, morning spacecraft series -- ASTER, CERES, MISR, MODIS and MOPITT instruments
ANSI	American National Standards Institute
ASCII	American Standard Code for Information Exchange
AT	acceptance test
ATT	acceptance test team
CCSDS	Consultative Committee for Space Data Systems
CDR	Critical Design Review
CDRL	contract data requirements list
CERES	Clouds and Earth's Radiant Energy System
CI	configuration item
CM	configuration management
COTS	commercial off-the-shelf (hardware or software)
CPU	central processing unit
CSC	computer software component
CSCI	computer software configuration item
CSMS	Communications and Systems Management Segment (ECS)
CSR	consent to ship review
CSU	computer software unit
DAA	data availability acknowledgment
DAAC	Distributed Active Archive Center
DADS	Data Archive and Distribution System
DAS	detailed activity schedule
DAN	data availability notice
DDA	data delivery acknowledgment
DDN	data delivery notice

DDTS	Distributed Defect Tracking System
DPR	Data Processing Request
DR	discrepancy report
ECS	EOSDIS Core System
EDF	ECS Development Facility
ESDIS	Earth Science Data and Information System (GSFC)
ETR	Element Test Review
F&PRS	Functional and Performance Requirements Specification
ftp	file transfer protocol [12/15/94]
GSFC	Goddard Space Flight Center
GUI	graphic user interface
H/W	hardware
HMI	human machine interface
HTML	HyperText Markup Language
HTTP	Hypertext Transport Protocol
HWCI	hardware configuration item
IDR	Incremental Design Review
I&AT	Integration and Acceptance Test
I&T	integration and test
I/F	interface
IATO	Independent Acceptance Test Organization
IR	interim release
IR-1	interim release-1
IV&V	independent verification and validation
L0-L4	Level 0 (zero) through Level 4
LAN	local area network
LaRC	Langley Research Center (DAAC)
LIS	Lightning Imaging Sensor
M&O	maintenance and operations
MSFC	Marshall Space Flight Center
NASA	National Aeronautics and Space Administration

NRCA	nonconformance reporting and corrective action
PDR	Preliminary Design Review
PGE	product generation executable
QA	quality assurance
R1	Release 1
SCF	Science Computing Facility
SDPF	Sensor Data Processing Facility (GSFC)
SDPS	Science Data Processing Segment (ECS)
SDR	Software Design Review
SDR	System Design Review
SFDU	Standard Format Data Unit
SMF	status message tool
SRR	System Requirements Review
TCP/IP	Transmission Control Protocol/Internet Protocol
TRMM	Tropical Rainfall Measuring Mission (joint US-Japan)
TRR	Test Readiness Review
TSDIS	TRMM Science Data and Information System
URL	universal reference location
V&V	verification and validation
WAIS	Wide Area Information Server
WWW	World-Wide Web
HP	Hewlett Packard
SGI	Silicon Graphics International
SUN	Sun MicroSystems

This page intentionally left blank.